EPSON
EXCEED YOUR VISION

ePOS-Print XML Ver.1.2 User's Manual

Overview

Describes the features and development environment.

Sample Program

Describes how to use the sample program and how to build a system.

Programming Guide

Describes how to write programs in Web application development.

ePOS-Print XML Reference

Describes the ePOS-Print XML.

Appendix

Describes the specifications for printers used for ePOS-Print.

M00048200
Rev.A

Cautions

- No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Seiko Epson Corporation.
- The contents of this document are subject to change without notice. Please contact us for the latest information.
- While every precaution has taken in the preparation of this document, Seiko Epson Corporation assumes no responsibility for errors or omissions.
- Neither is any liability assumed for damages resulting from the use of the information contained herein.
- Neither Seiko Epson Corporation nor its affiliates shall be liable to the purchaser of this product or third parties for damages, losses, costs, or expenses incurred by the purchaser or third parties as a result of: accident, misuse, or abuse of this product or unauthorized modifications, repairs, or alterations to this product, or (excluding the U.S.) failure to strictly comply with Seiko Epson Corporation's operating and maintenance instructions.
- Seiko Epson Corporation shall not be liable against any damages or problems arising from the use of any options or any consumable products other than those designated as Original EPSON Products or EPSON Approved Products by Seiko Epson Corporation.

Trademarks

EPSON® and ESC/POS® are registered trademarks of Seiko Epson Corporation in the U.S. and other countries.

Windows® and Internet Explorer® are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Android™ and Google Chrome™ are either registered trademarks or trademarks of Google Inc. in the United States and other countries.

Apple®, Mac®, Mac OS® and Xcode® are either registered trademarks or trademarks of Apple Inc. in the United States and other countries.

IOS® is registered trademarks or trademarks of Cisco in the United States and other countries.

Flash®, Flex™ are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and other countries.

Java™ is a registered trademark of Oracle Corporation, its subsidiaries, and affiliates in the U.S. and other countries.

Eclipse® is a trademark or registered trademark of Eclipse Foundation, Inc.

ESC/POS® Command System

EPSON has been taking industry's initiatives with its own POS printer command system (ESC/POS). ESC/POS has a large number of commands including patented ones. Its high scalability enables users to build versatile POS systems. The system is compatible with all types of EPSON POS printers (excluding the TM-C100) and displays. Moreover, its flexibility makes it easy to upgrade the future. The functionality and the user-friendliness is valued around the world.

Copyright © 2012 Seiko Epson Corporation. All rights reserved.

For Safety

Key to Symbols

The symbols in this manual are identified by their level of importance, as defined below. Read the following carefully before handling the product.

	Provides information that must be observed to avoid damage to your equipment or a malfunction.
	Provides important information and useful tips.

Restriction of Use

When this product is used for applications requiring high reliability/safety such as transportation devices related to aviation, rail, marine, automotive etc.; disaster prevention devices; various safety devices etc; or functional/precision devices etc, you should use this product only after giving consideration to including fail-safes and redundancies into your design to maintain safety and total system reliability. Because this product was not intended for use in applications requiring extremely high reliability/safety such as aerospace equipment, main communication equipment, nuclear power control equipment, or medical equipment related to direct medical care etc, please make your own judgment on this product's suitability after a full evaluation.

About this Manual

Aim of the Manual

This manual is aimed to provide developers with information required for developing and designing applications using ePOS-Print XML.

Manual Content

The manual is made up of the following sections:

Chapter 1	Overview
Chapter 2	Sample Program
Chapter 3	Programming Guide
Chapter 4	ePOS-Print XML Reference
Appendix	Printer specifications

Contents

■ **For Safety** **3**

Key to Symbols 3

■ **Restriction of Use** **3**

■ **About this Manual** **4**

Aim of the Manual 4
Manual Content 4

■ **Contents** **5**

Overview 9

■ **ePOS-Print XML** **9**

Features 9
Function 10
ePOS-Print Service 10
XML Schemas 11
Namespace 11
Print Image 11

■ **Operating Environment** **12**

Application Environment 12
Device 12
Development Environment 12
TM Intelligent Printer 12
TM Printers That Can Be Controlled 12

■ **Contents in the Package** **13**

Download 13

■ **Restrictions** **14**

Sample Program 15

■ **Overview** **15**

Sample Program Overview 15

■ **Environment** **16**

■ **Operating Environment** **17**

■ **Environment Settings** **18**

Network Settings for the TM Intelligent Printer 19
Device ID Settings 20
Editing the Sample Program 21
Registering the Sample Program 22

■ **Environment Settings in Other Execution Environments** **23**
HTML5 (To Build a Web Server on Windows/Linux) 23
Java 24
Windows (Visual Basic .NET) 25
Silverlight 26
Flash 27
Android 28
Windows CE 30
iOS 34

Programming Guide 35

■ **How to Use XML Schemas** **35**
Visual Studio 35
Eclipse 39
■ **ePOS-Print XML** **40**
Print Mode 40
Programming Flow 40
Print Document Creation 41
ePOS-Print XML Generator Tool 45
Transmission of Print Document 47
Reception of Print Result 48
Checking the Printer Status 51
■ **How to Create Graphic Data** **52**
How to Use 52
■ **Printing in Other Environments** **53**
Microsoft Silverlight 53
Adobe Flex 54

ePOS-Print XML Reference 55

■ **Element List** **55**
■ **Root Element** **56**
<epos-print> 56
<response> 57

■ **Element** **59**

<text> 59
<feed> 64
<image> 65
<logo> 67
<barcode> 68
<symbol> 73
<hline> 77
<vline-begin> 78
<vline-end> 79
<page> 80
<area> 81
<direction> 83
<position> 84
<line> 85
<rectangle> 86
<cut> 87
<pulse> 88
<sound> 89
<command> 90

Appendix **91**

■ **Printer specifications** **91**

TM-T88V-i 91
TM-T88V 93
TM-T70-i 95
TM-T70 96
TM-T90 97

Overview

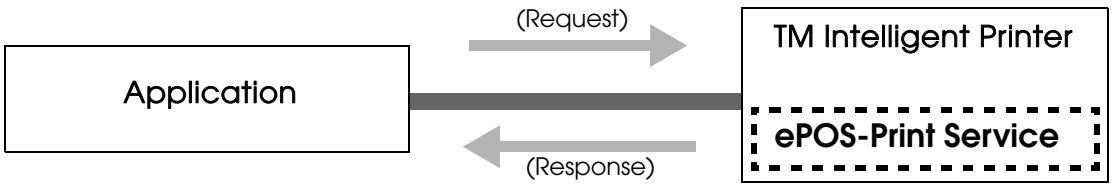
This chapter describes the features of and the specifications for ePOS-Print XML.

ePOS-Print XML

ePOS-Print XML is a POS printer command system that defines the major features of the ESC/POS command system using XML.

ePOS-Print XML allows printing to TM intelligent printers from applications in an environment or on OS where HTTP communication is possible. ePOS-Print XML embedded into native applications on smart phones and tablet terminals makes it possible to print from such devices.

When a print document is sent from a host to a TM intelligent printer using SOAP/HTTP, the TM intelligent printer performs printing and returns a response document.



Features

- ❏ Allows printing from SOAP/HTTP-supported devices. Not OS-dependent.
- ❏ Installation of drivers and plug-ins is not required.
- ❏ No PCs or servers are required for printing.
- ❏ Allows printing by TM printers via TM intelligent printers.
- ❏ Automatically checks the status of the TM printer before printing. There is no need for checking the status of the TM printer in advance.
- ❏ Does not respond to a printer's function to automatically send its status (AutoStatusBack). Instead, capable of sending an empty print command and checking the status of the TM printer based on the result of command transmission.
- ❏ To change the printer settings, utility programs dedicated to each printer or other utility programs should be used.
- ❏ Capable of obtaining the printer status even when the printer is offline.

Function

Item	Description
Common	Alignment, Rotation, Line Spacing, Page mode
Character	ANK
Character style	Font, Smoothing, Double-width, Double-height , Horizontal scale, Vertical scale, White/Black Reverse, Underlin, Emphasized Color, Print position
Paper feed	By Unit, By Line
Raster image	Mono, Multiple-tone (for TM-T88V-i and TM-T88V only)
NV logo	Mono, Multiple-tone (for TM-T88V-i and TM-T88V only)
Barcode	UPC-A, UPC-E, EAN13(JAN13), EAN8(JAN13), CODE39, ITF,CODABAR(NW-7), CODE93, CODE128, GS1-128, GS1
Two-Dimensional Code	PDF417, QR Code, MaxiCode, GS1 DataBar
Line	6 types
Paper cut	No Feed, Feed Cut, Cut Reservation
Drawer Kick-out	Connector, ON Time
Buzzer	7 Pattern, Repeat
Command	ESC/POS insertion
Response	Print result, Error code, Status

ePOS-Print Service

ePOS-Print Service is an XML Web service for printing using ePOS-Print XML. It is installed in TM intelligent printers.

When a print document is sent from a host to ePOS-Print Service using SOAP/HTTP, ePOS-Print Service performs printing and returns a response document.

Specification

Item	Description
Request message	A SOAP message is sent against a HTTP POST method request. One print document is specified in the SOAP body.
Response message	A SOAP message is returned against a HTTP POST method response. A response document is put in the SOAP body.

XML Schemas

The ePOS-Print XML syntax is defined by W3C XML Schemas.

XML schemas help applications strictly validate XML documents using the XML API of the operating environment. In addition, application developers can efficiently create XML documents using the XML document editing support function of the integrated development environment.

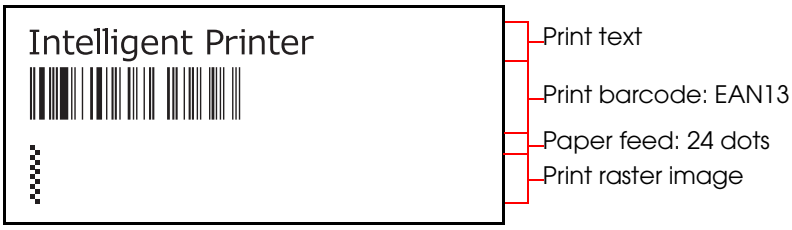
File name : epos-print-1.2.x.xsd

Namespace

The ePOS-Print XML namespaces and namespace identifiers used in this document are shown as follows:

Identifier	Namespace
(Default)	http://www.epson-pos.com/schemas/2011/03/epos-print
tns	http://www.epson-pos.com/schemas/2011/03/epos-print
xs	http://www.w3.org/2001/XMLSchema

Print Image



Operating Environment

Application Environment

An environment in which HTTP communication is possible and XML documents can be handled

- ❏ OS (Example: iOS/ Android/ Windows CE/ Windows/ Linux/ MacOS, etc.)
- ❏ HTML5-supported browser, Flex, Silverlight
- ❏ Java, .NET Framework

Device

A device capable of HTTP communication by connecting to the network.
(Example: smart phones, tablet terminals, computers, etc.)

Development Environment

An editor program capable of editing XML documents.
(Example: Visual Studio/Eclipse, etc.)

TM Intelligent Printer

- ❏ TM-T88V-i
- ❏ TM-T70-i

TM Printers That Can Be Controlled

- ❏ TM-T88V
- ❏ TM-T70
- ❏ TM-T90

Contents in the Package

Manual

- ❏ ePOS-Print XML User's Manual (This Document)
- ❏ TM-T88V-i Technical Reference Guide
- ❏ TM-T70-i Technical Reference Guide

Sample Program

ePOS-Print_XML_UM_E_Sample.zip

Contains the following:

- ❏ epos-print-1.2.x.xsd (ePOS-Print XML schemas)
- ❏ Sample program
  (HTML5/ Java/ Windows/ Silverlight/ Flash/Android/ iOS/ Windows CE)
- ❏ ePOS-Print XML Generator Tool

Utility

Utility	TM-T88V-i	TM-T70-i
TM-T88V Utility	●	-
TM Flash Logo Setup Utility	-	●
TMNet WinConfig	●	●

Download

For customers in North America, go to the following web site:

http://www.epsonexpert.com/ and follow the on-screen instructions.

For customers in other countries, go to the following web site:

https://download.epson-biz.com/?service=pos

Restrictions

- ❏ The drawer and the buzzer cannot be used together.
- ❏ The buzzer function cannot be used if the printer is not provided with the buzzer.
- ❏ When multiple tones are set for raster images, intermitting printing may occur because the amount of data to print increases and white stripes may appear in the print result.
- ❏ The scan quality of barcodes/two-dimensional symbols printed as multiple-tone raster images cannot be guaranteed. Print them as two-tone images.

Sample Program

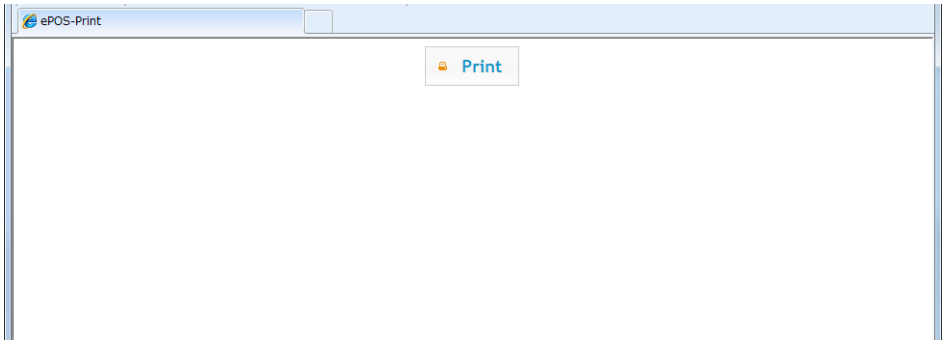
This chapter describes how to use the sample program.

> In this chapter, descriptions are made mainly based on a HTML5 environment. For an environment other than the above, read such descriptions in ways that suit the relevant environment.

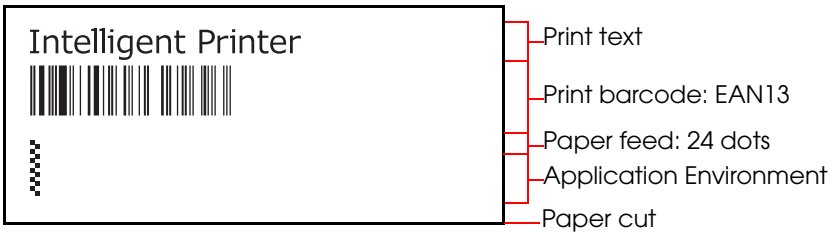
Overview

Sample Program Overview

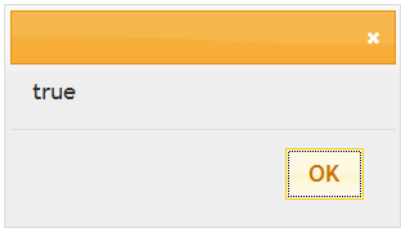
The following screen appears when the sample program starts. Click the (Print) button to execute.



Sample program execution result



A response message from the TM intelligent printer



Environment

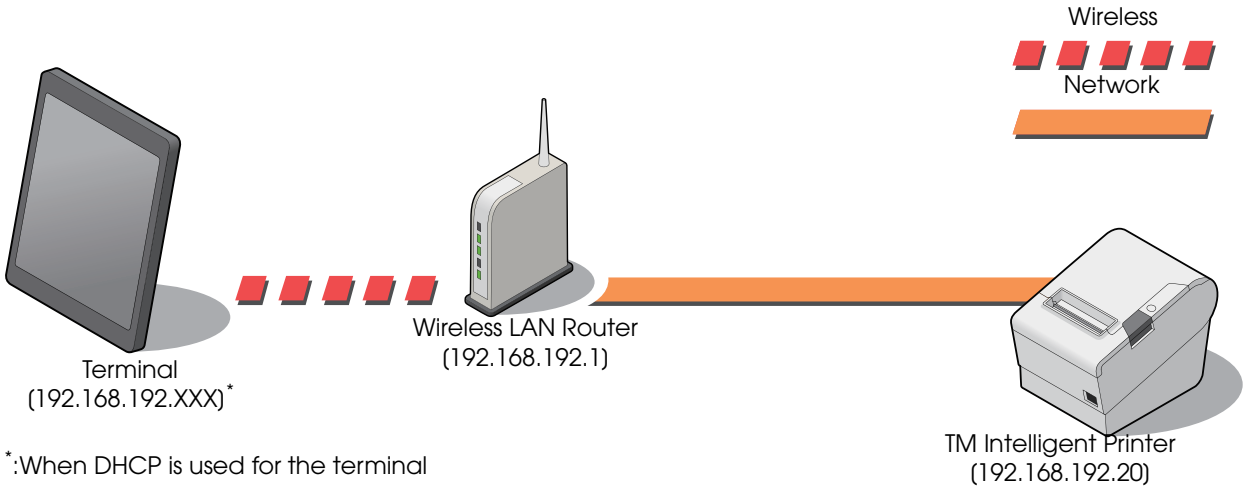
This section describes the following system: the device's HTML 5-suported browser accesses the sample program uploaded to the TM intelligent printer Web server and then printing is performed. For use in other environments, see "Environment Settings in Other Execution Environments" on page 23.
The following environments are provided for the sample program:

- HTML5-supported Web browser
- Java
- Windows (Visual Basic .NET)
- Silverlight
- Flash
- Android
- iOS
- Windows CE

Operating Environment

The system configuration diagram for the sample programs is as below.

> The figure below also describes an example of IP address settings as network settings.



*:When DHCP is used for the terminal

- ❏ Wireless LAN Router
- ❏ TM Intelligent Printer (1 set)
  TM-T88V-i/ TM-T70-i
- ❏ Terminal with an HTML5-supported Web browser

Environment Settings

A flow for configuring the environment settings for the sample program is shown as follows:

1. Router Settings

Configure the settings such as SSID, IP address, DHCP, and allocated IP address. For details, refer to the manual for the device you are using.

2. Network Settings for the TM Intelligent Printer (p.19)

Configuration is done from a Web browser.

3. Network settings for the printer to be controlled

Configure these settings by using the network setting utility. For details, refer to the detailed instruction manual for the printer (these settings are not required in this sample program).

4. Terminal settings

Configure the wireless LAN (Wi-Fi) settings of the terminal to match the router settings so as to enable network connection. For details, refer to the manual for the device you are using.

5. Device ID Settings (p.20)

Configuration is done from a Web browser(these settings are not required in this sample program).

6. Editing the Sample Program (p.21)

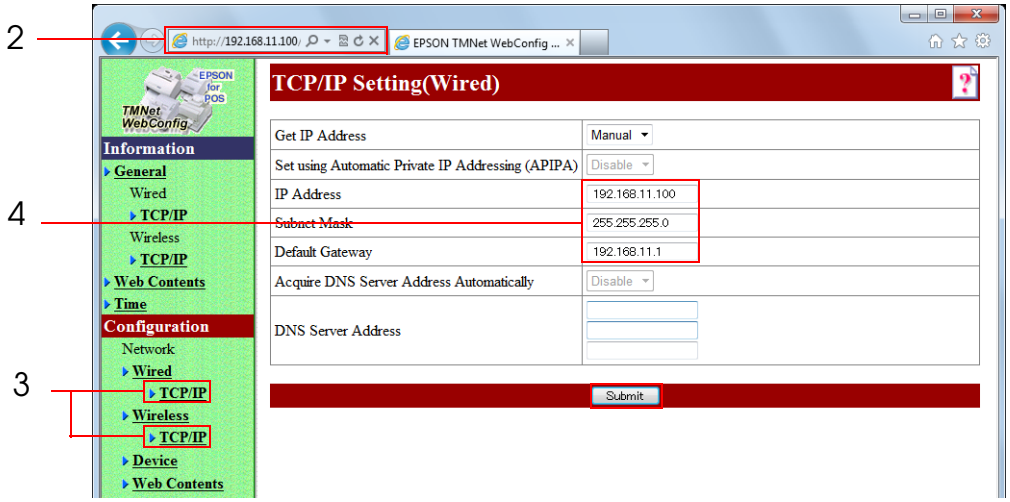
Write the IP address of the TM intelligent printer in the source code of the sample program using an editor program.

7. Registering the Sample Program (p.22)

Make registration using a Web browser.

Network Settings for the TM Intelligent Printer

Use TMNet WebConfig to configure the network settings such as IP address for the printer.



Configure the settings according to the following procedure:

1 Connect the printer to the network and turn the power ON.

2 Start the Web browser and type the URL of the TM intelligent printer interface into the address bar. Enter the password.

http://(IP address of the TM intelligent printer)/PrinterConfigurationPage/

> The login name for the TM intelligent printer is "epson", and the default password is "epson".

3 TMNet WebConfig starts.
Select as (Configuration) - (Wired / Wireless) - (TCP/IP).

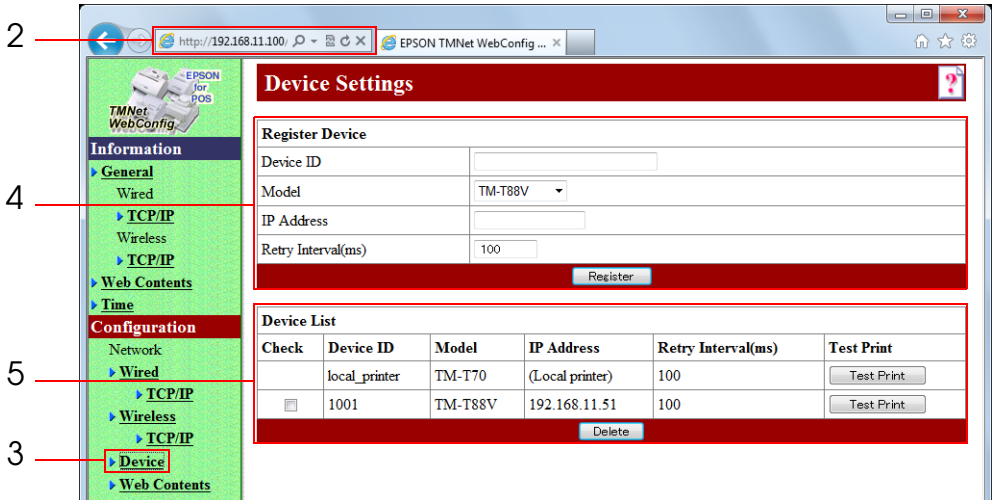
4 The "TCP/IP Setting" screen appears.
Configure the network settings for the TM intelligent printer and click (Submit).

5 Print the status sheet using the TM intelligent printer to check that the IP address has been updated.

Device ID Settings

Set the Device ID of the printer to be controlled by ePOS-Print into the TM intelligent printer. Use TMNet WebConfig to set the Device ID.

> In the sample program, "Device ID Settings" are not required. Refer to it if necessary.



Configure the settings according to the following procedure:

1. Connect all the printers to the network and turn their power ON.

2. Start the Web browser and enter the IP address set in "Network Settings for the TM Intelligent Printer" on page 19.

3. TMNet WebConfig starts.
   Select as (Configuration)-(Device).

4. The "Device Settings" screen appears. Set the following and click (Register).

Item	Description
Device ID	Specifies the ID to identify the printer to be controlled by ePOS-Print.
Model	Specifies the model of the printer to be controlled.
IP Address	Specifies the IP address of the printer to be controlled.
Retry Interval (ms)	Specifies the interval of retry toward the printer to be controlled, in milliseconds.

5. Information on the registered devices is displayed in (Device List).
   Click (Test Print) for each registered printer to check that it operates correctly.

Editing the Sample Program

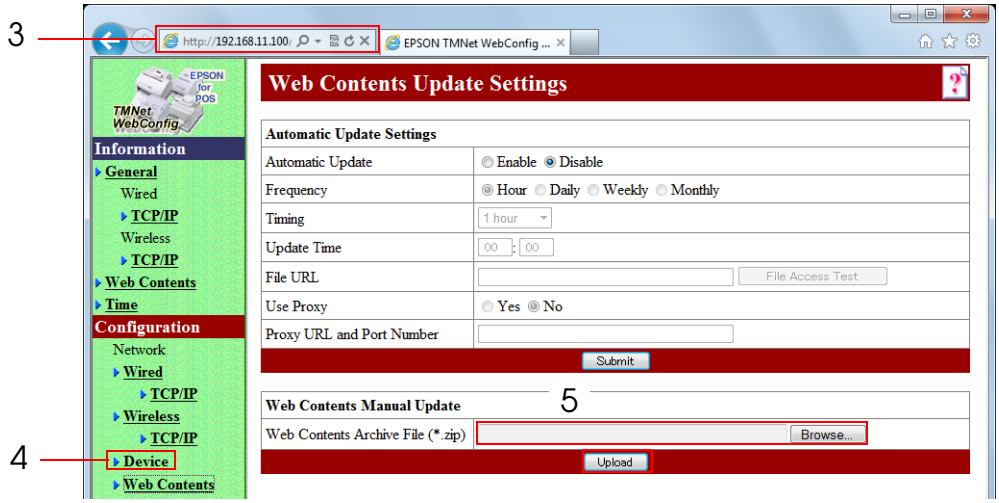
Write the IP address of the TM intelligent printer in the sample program. Edit the program as follows:

1. Expand HtmlApplication1.zip.
2. Open index.html using an editor program.
3. Change the source code portion "// URL of intelligent printer" in the source file to the IP address of the TM intelligent printer and the save the source file.

```
// URL of intelligent printer
var address = 'http://192.168.192.168/cgi-bin/epos/
              service.cgi?devid=local_printer&timeout=10000
```

Registering the Sample Program

Register the sample program into the TM intelligent printer.



Register as follows:

1 Compress the sample program to a zip file.
Put the application file (HtmlApplication1.zip) in the root folder of the creation environment.

```
HtmlApplication1.zip ---------- : zip file
 ├─css/
 ├─img/
 ├─js/
 └─index.html ---------- : Application file
```

2 Connect the printer to the network and turn the power ON.

3 Start the Web browser and type the IP address set in "Network Settings for the TM Intelligent Printer" on page 19.

4 TMNet WebConfig starts.
Select as (Configuration)-(Web Contents).

5 The "Web Contents Update Setting" screen appears. Click (Browse) and specify the compressed file created in step 1, and then click (Upload).

6 Start the web browser and enter the URL for the sample program.
http://(IP address of the TM intelligent printer)/index.html

Environment Settings in Other Execution Environments

This section describes the setting operation in a development environment up to sample program execution.

- ❏ HTML5 (To Build a Web Server on Windows/Linux) (p. 23)
- ❏ Java (p. 24)
- ❏ Windows (Visual Basic .NET) (p. 25)
- ❏ Silverlight (p. 26)
- ❏ Flash (p. 27)
- ❏ Android (p. 28)
- ❏ Windows CE (p. 30)
- ❏ iOS (p. 34)

HTML5 (To Build a Web Server on Windows/Linux)

Item	Description
Development environment (example)	Microsoft Visual Studio 2010
Required item	IIS (Internet Information Services) or Apache
Sample program file name	HtmlApplication1.zip
Source editing	Edit the IP address of the TM intelligent printer to suit the network environment. The default value for the IP address of the TM intelligent printer is "**http://192.168.192.168**".

Environment Setting Procedure

1. Install IIS.
2. Configure your computer and TM intelligent printer so that they can connect to the network.
3. Uncompress the sample program and copy the uncompressed sample program files into the root folder of IIS.
   System drive : **\Inetpub\wwwroot**
4. Open the sample program html file using Visual Studio.

Java

Item	Description
Development environment (example)	Eclipse
Required item	Java SE Development Kit 6
Sample program file name	JavaProject1.zip
Source editing	Edit the IP address of the TM intelligent printer to suit the network environment. The default value for the IP address of the TM intelligent printer is "**http://192.168.192.168**".

Environment Setting Procedure

1. Obtain Java SE 6 and install that program.
2. Configure your computer and TM intelligent printer so that they can connect to the network.
3. Uncompress the sample program into any folder.
4. Import the sample program using Eclipse.

Windows (Visual Basic .NET)

Item	Description
Development environment (example)	Microsoft Visual Studio 2010
Required item	-
Sample program file name	WindowsApplication1.zip
Source editing	Edit the IP address of the TM intelligent printer to suit the network environment. The default value for the IP address of the TM intelligent printer is "**http://192.168.192.168**".

Environment Setting Procedure

1. Configure your computer and TM intelligent printer so that they can connect to the network.
2. Uncompress the sample program into any folder.
3. Open the sample program solution file using Visual Studio.

Silverlight

Item	Description
Development environment (example)	Microsoft Visual Studio 2010
Required item	Microsoft Silverlight 4 Tools for Visual Studio 2010
Sample program file name	SilverlightApplication1.zip, crossdomain.zip
Source editing	Edit the IP address of the TM intelligent printer to suit the network environment. The default value for the IP address of the TM intelligent printer is "**http://192.168.192.168**".

Environment Setting Procedure

1. Register the cross domain policy file in the TM intelligent printer. Refer to Registering the Sample Program (p. 22) to upload "crossdomain.zip".
2. Download Microsoft Silverlight 4 Tools for Visual Studio 2010(Silverlight4_Tools.exe) and install that program.
3. Configure your computer and TM intelligent printer so that they can connect to the network.
4. Uncompress the sample program into any folder.
5. Open the sample program solution file using Visual Studio.

Flash

Item	Description
Development environment	Flash Builder 4.5 or later
Required item	Flash Builder 4.5 or later
Sample program file name	FlexProject1.zip, crossdomain.zip
Source editing	Edit the IP address of the TM intelligent printer to suit the network environment. The default value for the IP address of the TM intelligent printer is "**http://192.168.192.168**".

Environment Setting Procedure

1. Register the cross domain policy file in the TM intelligent printer. Refer to Registering the Sample Program (p. 22) to upload "crossdomain.zip".
2. Download Adobe Flash Builder from Flash Platform.
3. Configure your computer and TM intelligent printer so that they can connect to the network.
4. Uncompress the sample program into any folder.
5. Start Adobe Flash Builder and read the sample program project.

Android

Item	Description
Development environment	Eclipse
Required item	• Android SDK 1.5 • Java SE Development Kit 6 • ADT(Android Development Tools) Plugin for Eclipse
Sample program file name	AndroidProject1.zip
Source editing	Edit the IP address of the TM intelligent printer to suit the network environment. The default value for the IP address of the TM intelligent printer is "**http://192.168.192.168**".

Environment Setting Procedure

1. Obtain Android SDK and expand it into any folder.
2. Download Java SE Development Kit 6 and install that program.
3. Start Android SDK Manager and install the Android 1.5(API3) package.
4. Start Android AVD Manager and create (Android Virtual Device).
See "Creating Android Virtual Device" on page 29.
5. Install ADT Plugin in Eclipse.
See "Installing ADT Plugin" on page 29
6. Change the settings of Eclipse.
See "Changing the settings of Eclipse" on page 29
7. Configure your computer and TM intelligent printer so that they can connect to the network.
8. Uncompress the sample program into any folder.
9. Import the sample program using Eclipse.

Creating Android Virtual Device

1. Start Eclipse.
2. Select as (Windows)-(Android SDK and AVD Manager).
3. The "Android SDK and AVD Manager" screen appears. Select the (New) button.
4. On (Create new Android Virtual Device), type the name. Alternatively, select the device to use and create a new entry.

Installing ADT Plugin

1. Start Eclipse.
2. From the menu items, select as (Help)-(Install New Software).
3. The "Install" screen appears. Click the (Add) button.
4. Enter values in the (Name) and (Location) fields and click the (OK) button.

Item	Description
Name	Enter any name.
Location	Enter (https://dl-ssl.google.com/android/eclipse/).

5. Return to the "Install" screen. Tick (Developer Tools) and click the (Next) button.
6. When the installation process finishes, restart Eclipse.

Changing the settings of Eclipse

1. Start Eclipse.
2. Select as (Windows)-(Configuration).
3. The "Setting" screen appears. Select (Android).
   Although a warning dialog box appears, telling you that a path to Android SDK has not been set, there is no problem.
4. Click the (Browse) button.
5. Select the (Android - Android SDK) directory and click the (OK) button.

Windows CE

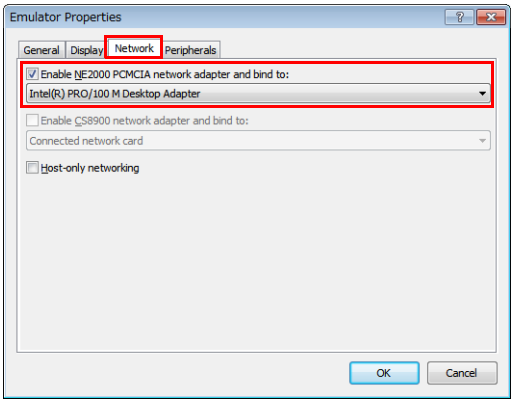
Item	Description
Development environment	Microsoft Visual Studio 2008 SP1
Required item	• Windows Embedded Compact 7 • Virtual PC 2007
Sample program file name	SmartDeviceProject1.zip
Source editing	Edit the IP address of the TM intelligent printer to suit the network environment. The default value for the IP address of the TM intelligent printer is "**http://192.168.192.168**".

Environment Setting Procedure

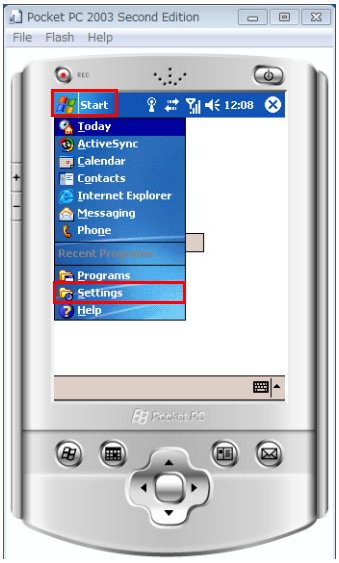
1. Install Visual Studio 2008 SP1.
2. Download Windows Embedded Compact 7 and install that program.
3. Install Virtual PC 2007.
4. Configure your computer and TM intelligent printer so that they can connect to the network.
5. Uncompress the sample program into any folder.
6. Open the sample program solution file using Visual Studio.
7. Execute the sample program using Visual Studio.

8 The Device Emulator starts. In the Device Emulator, select as (File)-(Configure...)-(Network).

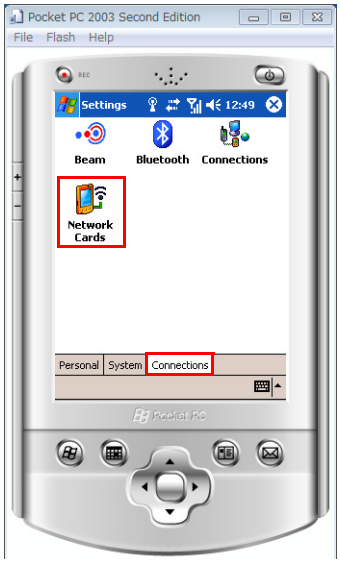
9 The "Emulator Properties" dialog box appears. Select (Network).
Tick "Enable NE2000 PCMCIA network adapter and bind to:" and select the computer's network card. Click the (OK) button.



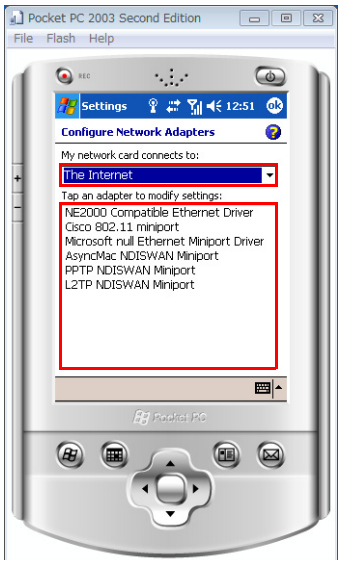
10 In the Device Emulator, select as (Start)-(Settings).



11 In the Device Emulator, select as (Settings)-(Connections)-(Network Cards).

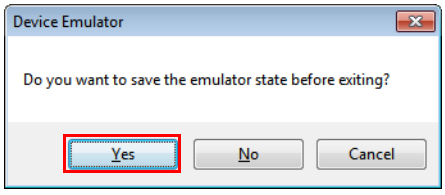


12 In the Device Emulator, select (The Internet) and then select the same network card as that of your computer.



13 Set the IP address and click (OK) at the top right of the screen.

14 A confirmation dialog box appears. Click (Yes).



15 Restart the Device Emulator.

iOS

Item	Description
OS	Mac OS X 10.6.8
Development environment	Apple Xcode 3.2.6
SDK	iOS SDK 4.3
Sample program file name	iOSDeviceProject1.zip
Source editing	Edit the IP address of the TM intelligent printer to suit the network environment. The default value for the IP address of the TM intelligent printer is "**http://192.168.192.168**".
Execution	The sample program can be checked using iOS Simulator.

Environment Setting Procedure

1. Expand iOSDeviceProject1.zip.
2. Start XCode and open the (iOSDeviceProject1.xcodeproj) project file.
3. When the project is opened, the project settings and source files will be displayed.
4. From the options displayed when (Set Active Executable) is selected, select an iPhone or iPad file.

Programming Guide

This chapter describes how to write programs in the application development using ePOS-Print.

How to Use XML Schemas

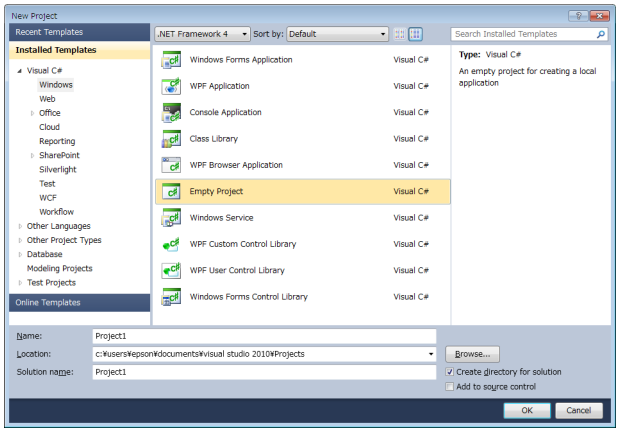
The XML document editing support function in the integrated development environment helps efficiently create XML documents and validate XML documents.

This section describes specific procedures using Visual Studio (p. 35) and Eclipse (p. 39) as examples.

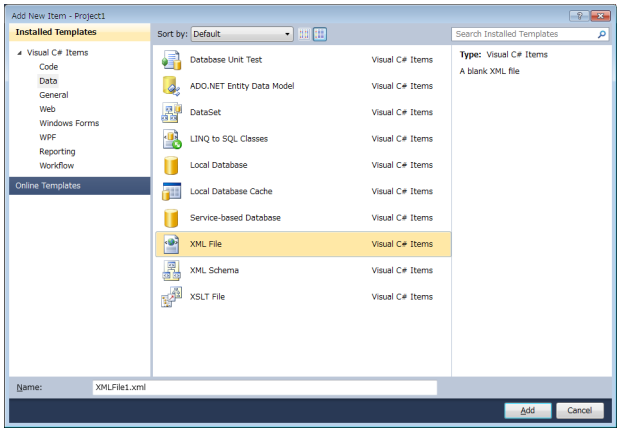
Visual Studio

Setting Procedure

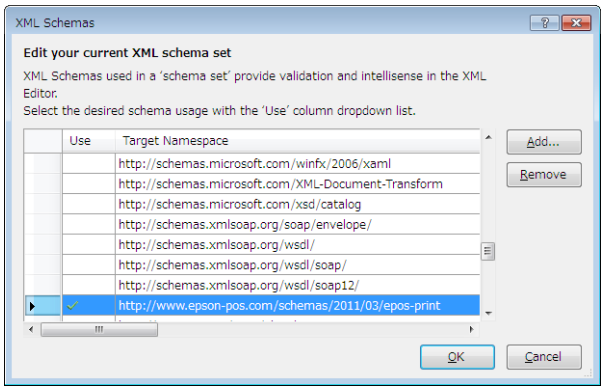
1 Create a new project.
(File)-(New)-(Project...)



2 Add an XML file to the created project.
(Project)-(Add New Item...)



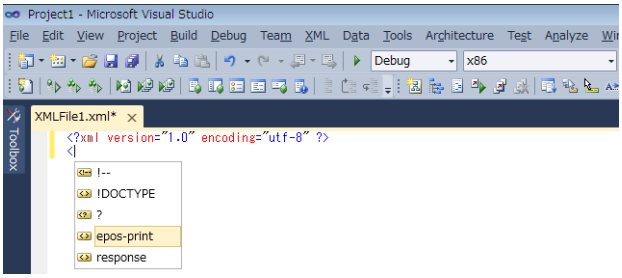
3 Set the XML schema "epos-print-1.2.x.xsd" to the XML file.
(XML)-(Schemas...)-(Add...)



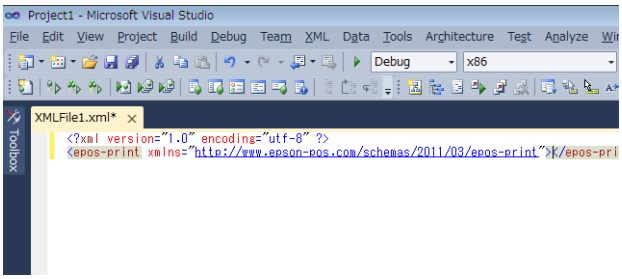
Editing Example

For details on elements, refer to ePOS-Print XML Reference (p.55).

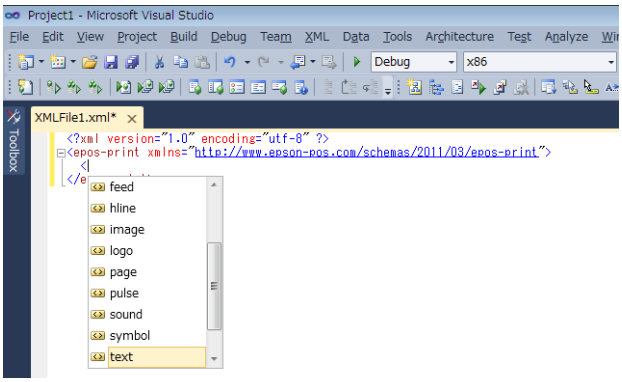
1 Enter the character "<" in the XML file. Then, root element candidates appear.



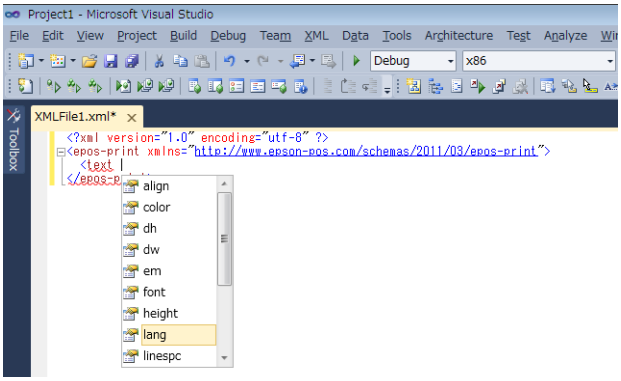
2 Select "epos-print" from the candidates and enter the character ">". Then, the epos-print root element is added.



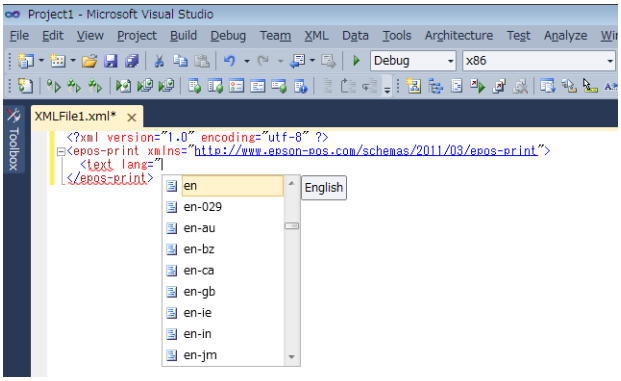
3 Start a new line and enter the character "<". Then, element candidates appear.



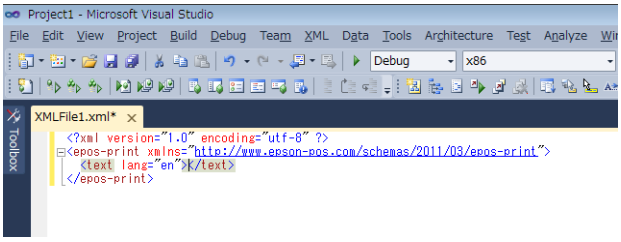
4 Select "text" from the candidates and enter an empty space. Then, attribute candidates appear.



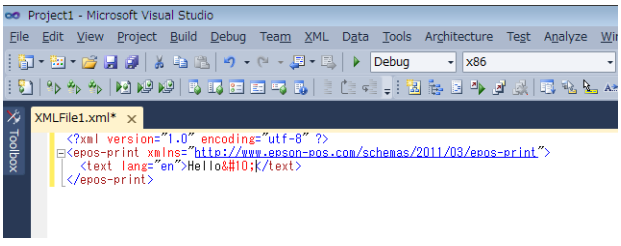
5 Select "lang" from the candidates. Then, attribute value candidates appear.



6 Select "en" from the candidates and enter the character ">". Then, the text element is added.



7 Enter a character string to print for the content of the text element.



Eclipse

Setting Procedure

Create an XML file and set the location of the schema using URL as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<epos-print xmlns:xsi=”http://www.w3.org/2001/XMLschema-instance”
xsi:schemaLocation="http://www.epson-pos.com/schemas/2011/03/epos-print
epos-print-1.2.x.xsd"
xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">

<text lang='en'>Hello&#10;</text>

</epos-print>
```

For details on elements, refer to ePOS-Print XML Reference (p.55).

ePOS-Print XML

In this chapter, descriptions are made mainly based on a HTML5 environment. For an environment other than the above, read such descriptions in ways that suit the relevant environment.

Print Mode

There are two types of print modes: standard and page modes.

Standard mode

In standard mode, characters are printed line by line. The line feed space is adjusted based on the font size and the height of images, barcodes, etc. This mode is suitable for the type of printing such as printing receipts that requires the paper length to change according to the print space.

Page mode

In page mode, you set a print area, lay out data in it, and print the data in a batch operation. Characters, images, and barcodes are laid out in the print positions (coordinates).

Programming Flow

For the ePOS-Print XML, programming is performed based on the following work flow:

1. Print Document Creation (p 41)
   - ❏ To create a text print document: (p.42)
   - ❏ To create a graphic print document: (p.43)
   - ❏ To create a page mode print document: (p.44)

2. Transmission of Print Document (p 47)

3. Reception of Print Result (p 48)

To ensure successful print operation, write a program in such a way that data is sent after checking the printer status. For the above procedure, refer to Checking the Printer Status (p.51).

Print Document Creation

Create a print document using elements in the epos-print root element. For details on elements, refer to Element List (p.55).

Refer to the following program for print document creation.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>TITLE</title>
<script type="text/javascript">
  function createDocument() {
    //Start print document creation.
    var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">';
    //Create a print document
    request += '<text lang="en" smooth="true"/>';
    request += '<text font="font_a"/>';
    request += '<text width="3" height="3">Hello, World!&#10;</text>';
    request += '<cut type="feed"/>';
    //End print document creation.
    request += '</epos-print>';
  }
</script>
</head>
<body>
</body>
</html>
```

Create a print document

To create a text print document:

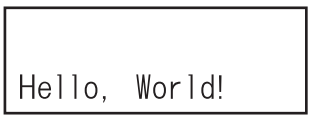
Create a text print document using the text element in the epos-print root element. Configure the settings for the text to print using the attributes of the text element.
Refer to the following program for print document creation.

For the string "Hello World!", to create a print document based on the following settings:

- Font: FontA
- Scale: x 4 (horizontal) and x 4 (vertical)
- Style: Bold

```
<script type="text/javascript">
  function createDocument() {
    //Start print document creation.
    var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/
epos-print">';
    //Create a print document
    //<Configure the print character settings>
    request += '<text lang="en"/>';
    request += '<text smooth="true"/>';
    request += '<text font="font_a"/>';
    request += '<text width="4" height="4"/>';
    request += '<text em="true"/>';
    //<Specify the character string to print>
    request += '<text>Hello, World!&#10;</text>';
    //<Specify the feed cut>
    request += '<cut type="feed"/>';
    //End print document creation.
    request += '</epos-print>';
  }
</script>
```

Print image



To create a graphic print document:

Create a graphic print document using the image element in the epos-print root element. Set the image size using an attribute of the image element. For graphics, specify raster graphic bit-image data using a base 64 encoded character string.
Refer to the following program for print document creation.

To create a print document with a checkered flag raster image:

```
<script type="text/javascript">
  function createDocument() {
    //Start print document creation.
    var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/
epos-print">';
    //Create a print document
    //<Specify the raster image>
    request += '<image width="8" height="48">8PDw8A8PDw/w8PDwDw8PD/
Dw8PAPDw8P8PDw8A8PDw/w8PDwDw8PD/Dw8PAPDw8P</image>';
    //<Specify the feed cut>
    request += '<cut type="feed"/>';
    //End print document creation.
    request += '</epos-print>';
  }
</script>
```

- This section describes how to print a raster image. In addition, there is also a method of printing graphics registered in the NV memory of the printer. For details, refer to <logo> (p.67).
- For details on how to create raster images, refer to How to Create Graphic Data (p.52).

Print image

To create a page mode print document:

Create a graphic print document using the image element in the epos-print root element. Set the image size using an attribute of the image element. For graphics, specify raster graphic bit-image data using a base 64 encoded character string.
Refer to the following program for print document creation.

For the string "Hello World!", to create a print document based on the following settings:

- Page mode print area (in dots)
  Origin of horizontal axis: 100, origin of vertical axis: 50, width: 200, height: 100
- Page mode print positions (in dots)
  Horizontal print position: 0, vertical print position: 42
- Font: FontA
- Scale: x 2 (horizontal) and x 2 (vertical)
- Style: Bold

```
<script type="text/javascript">
  function createDocument() {
    //Start print document creation.
    var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/
epos-print">';
   //Create a print document
   //<The page mode starts>
    request += '<page>';
   //<Specify the page mode print area>
    request += '<area x="100" y="50" width="200" height="100"/>';
   //<Specify the page mode print position>
    request += '<position x="0" y="42"/>';
    //<Set print text>
    request += '<text lang="en"/>';
    request += '<text smooth="true"/>';
    request += '<text font="font_a"/>';
    request += '<text width="2" height="2"/>';
    request += '<text em="true"/>';
    //<Specify the character string to print>
    request += '<text>Hello,</text>';
    request += '<position x="0" y="96"/>';
    request += '<text>World!</text>';
   //<The page mode ends>
    request += '</page>';
    //<Specify the feed cut>
    request += '<cut type="feed"/>';
    //End print document creation.
    request += '</epos-print>';
  }
</script>
```

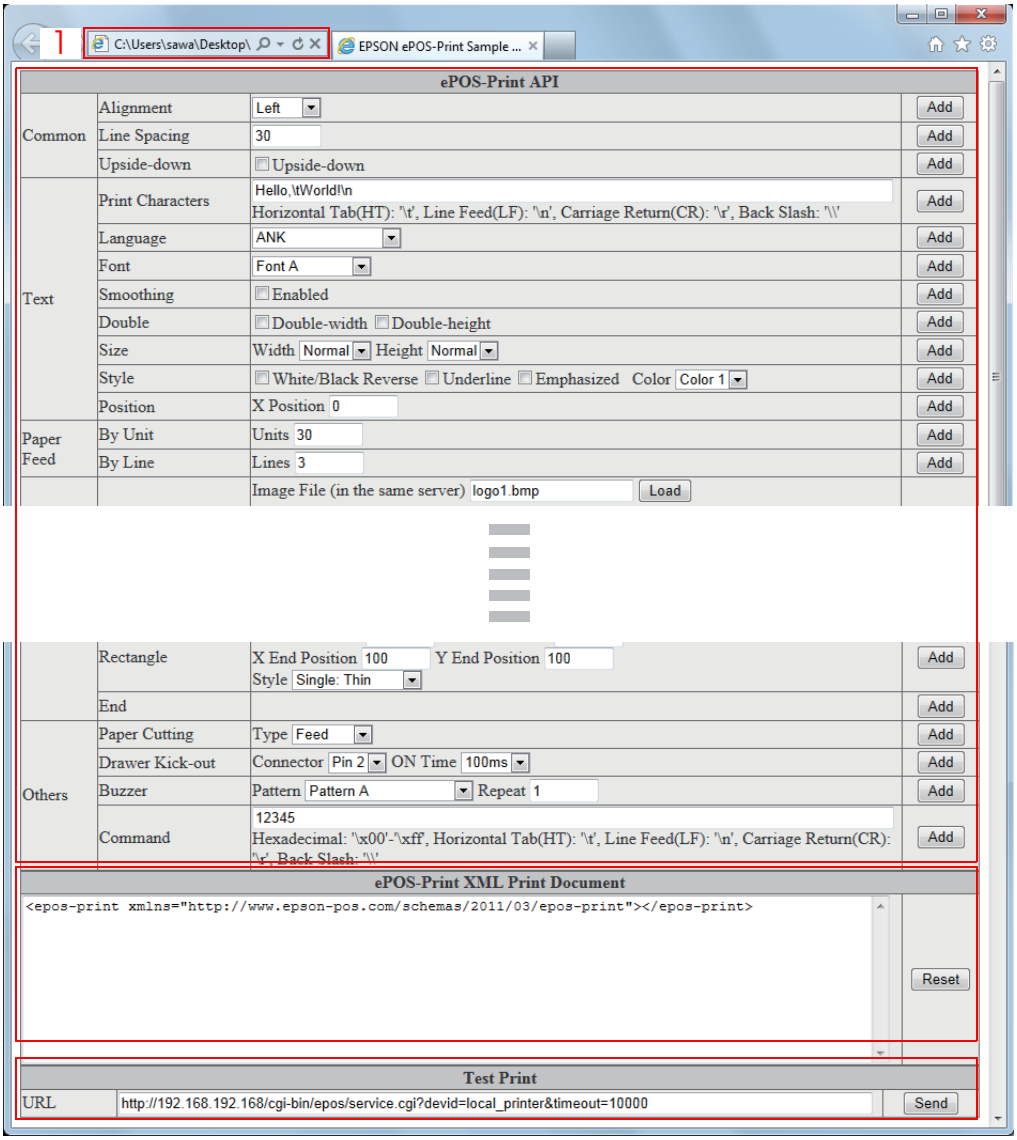
Print image

ePOS-Print XML Generator Tool

This section describes how to use the ePOS-Print XML generator tool included in the contents in the package.

This tool is a support tool for generating XML data. The tool helps you create ePOS-Print XML (p. 55) print document as you like. In addition, the generated XML data can be printed for testing. Use this tool for your application development.

How to Use This Tool



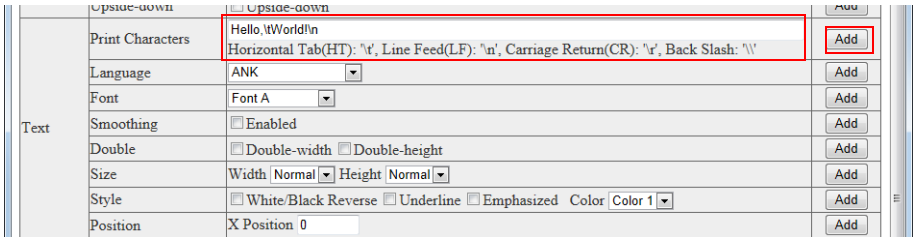
1 Copy the ePOS-Print XML generator tool file to the Web server and open the following URL using a Web browser.

http://(Web server IP address)/index.html

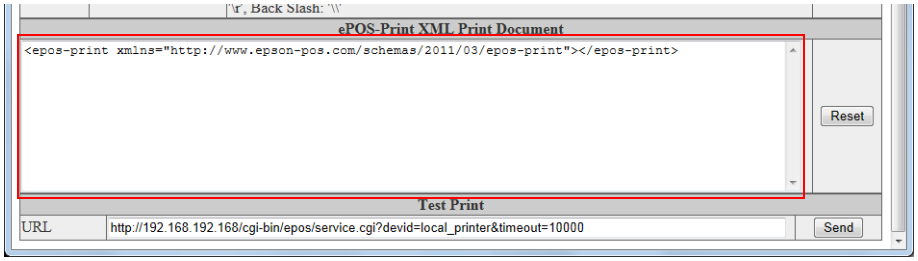
The ePOS-Print XML generator tool file is available in the folder to which ePOS-Print_XML_UM_E_Sample.zip was uncompressed (index.html, epos-print-1.2.x.js, logo1.bmp, logo2.bmp).

2 "EPSON ePOS-Print Sample Program" appears.

3 Set the ePOS-Print XML features and click the (Add) button.

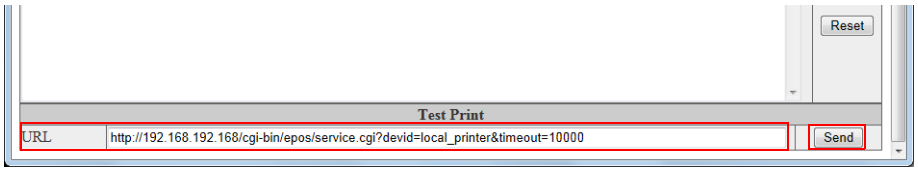


4 The source for the features added in step 3 is displayed on the "ePOS-Print XML Print Document" screen. It can be used by copying.

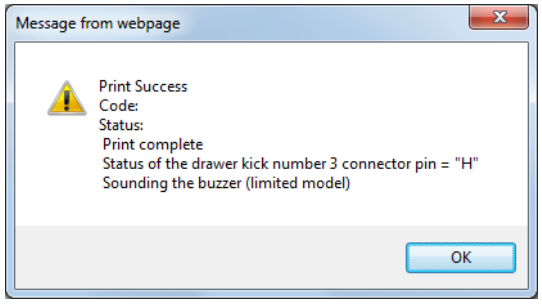


5 Enter the following URL in the (URL) field and click the (Send) button.

http://(the IP address of the TM intelligent printer)/cgi-bin/epos/service.cgi?devid=(the device ID of the printer used for printing)&timeout=(timeout time)



6 The print result will be displayed.



Transmission of Print Document

A print document is sent using an ePOS-Print object.
Create an ePOS-Print object using the constructor and specify the end point address for the printer to be used for printing as well as the print document into the send method to send the document.
For the details about the printer end point address, refer to Printer End Point Address (p.48).
Refer to the following program.

The TM intelligent printer checks the status of the TM printer used for printing and then start printing operation.

Refer to the following program.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>TITLE</title>
<script type="text/javascript">
  function sendDocument() {
    //Create a print document
    var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">';
    request += '<text lang="en" smooth="true"/>';
    request += '<text font="font_a"/>';
    request += '<text width="3" height="3">Hello, World!&#10;</text>';
    request += '<cut type="feed"/>';
    request += '</epos-print>';

    //Create a SOAP envelop
    var soap = '<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">' +
               '<s:Body>' + request + '</s:Body></s:Envelope>';

    //Create an XMLHttpRequest object
    var xhr = new XMLHttpRequest();
    //Set the end point address
    var url = 'http://192.168.192.168/cgi-bin/epos/
               service.cgi?devid=local_printer&timeout=10000';
    //Open an XMLHttpRequest object
    xhr.open('POST', url, true);
    //<Header settings>
    xhr.setRequestHeader('Content-Type', 'text/xml; charset=utf-8');
    xhr.setRequestHeader('If-Modified-Since', 'Thu, 01 Jan 1970 00:00:00 GMT');

    // Send print document
    xhr.send(soap);
  }
</script>
</head>
<body>
</body>
</html>
```

Send print document

Printer End Point Address

Specify the printer end point address in the following format:
http://(domain)/cgi-bin/epos/service.cgi?devid=(device ID)&timeout=(timeout time)

Items to specify	Description
Domain	Specify either the IP address or the domain name of the TM intelligent printer.
Device ID	Specifies the printer to be used for printing. Specify the Device ID registered using the TM intelligent printer's TMNet WebConfig.
Timeout period	Specifies the time to abort the process in milliseconds. The timeout parameter is optional; when it is omitted, 60 seconds (60000) is set. When the timeout period elapses, the print job is canceled; the data already interpreted by the printer before the start of the print abort process is printed.

Reception of Print Result

Set a callback function using the response root element (p. 57) to receive print results.
The following information is obtained:

- Print result
- Error code
- Printer status

The printer status can be obtained when communication with the printer is possible.

Refer to the following program. For the details about how to program a callback function in detail, refer to Error handling (p.50).

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>TITLE</title>
<script type="text/javascript">
  function sendDocument() {
    //Create a print document
    var request = '<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">';
    request += '<text lang="en" smooth="true"/>';
    request += '<text font="font_a"/>';
    request += '<text width="3" height="3">Hello, World!&#10;</text>';
    request += '<cut type="feed"/>';
    request += '</epos-print>';

    //Create a SOAP envelop
    var soap = '<?xml version="1.0" encoding="utf-8"?>' +
               '<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
               <s:Body>' + request + '</s:Body></s:Envelope>';

    //Create an XMLHttpRequest object
    var xhr = new XMLHttpRequest();
    //Set the end point address
    var url = 'http://192.168.192.168/cgi-bin/epos/
               service.cgi?devid=local_printer&timeout=10000';
    //Open an XMLHttpRequest object
    xhr.open('POST', url, true);
    //<Header settings>
    xhr.setRequestHeader('Content-Type', 'text/xml; charset=utf-8');
    xhr.setRequestHeader('If-Modified-Since', 'Thu, 01 Jan 1970 00:00:00 GMT');

    //Set a callback function
    xhr.onreadystatechange = function () {
        if (xhr.readyState == 4) {
            if (xhr.status == 200) {
                //Obtain the response root element
                var res = xhr.responseXML;
                var success = res.getElementsByTagName('response')[0].getAttribute('success');
                //When the printing is not successful, display a message
                if (!/^(1|true)$/.test(success)) {
                        alert('A print error occurred');
                }
            }
        }

    // Send print document
    xhr.send(soap);
  }
</script>
</head>
<body>
</body>
</html>
```

Callback function for print result reception

Error handling

Refer to the following program for the error handling method by a callback function.

```
// Set a response receipt callback function
xhr.onreadystatechange = function () {
                .
                .
                .
    // Obtain the print result and error code
    var res = xhr.responseXML;
    var msg = 'Print' + (res[0].getAttribute('success') ? 'Success' : 'Failure') +
              '\nCode:' + res[0].getAttribute('code') +
              '\nStatus:\n';
    // Obtain the printer status
    var asb = res[0].getAttribute('status');
    if (asb & 0x00000001) {
        msg += ' No printer response\n';
    }
    if (asb & 0x00000002) {
        msg += ' Print complete\n';
    }
    if (asb & 0x00000004) {
        msg += ' Status of the drawer kick number 3 connector pin = "H"\n';
    }
    if (asb & 0x00000008) {
        msg += ' Offline status\n';
    }
    if (asb & 0x00000020) {
        msg += ' Cover is open\n';
    }
    if (asb & 0x00000040) {
        msg += ' Paper feed switch is feeding paper\n';
    }
    if (asb & 0x00000100) {
        msg += ' Waiting for online recovery\n';
    }
    if (asb & 0x00000200) {
        msg += ' Panel switch is ON\n';
    }
    if (asb & 0x00000400) {
        msg += ' Mechanical error generated\n';
    }
    if (asb & 0x00000800) {
        msg += ' Auto cutter error generated\n';
    }
    if (asb & 0x00002000) {
        msg += ' Unrecoverable error generated\n';
    }
    if (asb & 0x00004000) {
        msg += ' Auto recovery error generated\n';
    }
    if (asb & 0x00020000) {
        msg += ' No paper in the roll paper near end detector\n';
    }
    if (asb & 0x00080000) {
        msg += ' No paper in the roll paper end detector\n';
    }
    if (asb & 0x80000000) {
        msg += ' Stop the spooler\n';
    }
    //Display in the dialog box
    alert(msg);
}
```

Checking the Printer Status

To check the printer status without printing, send empty print data.

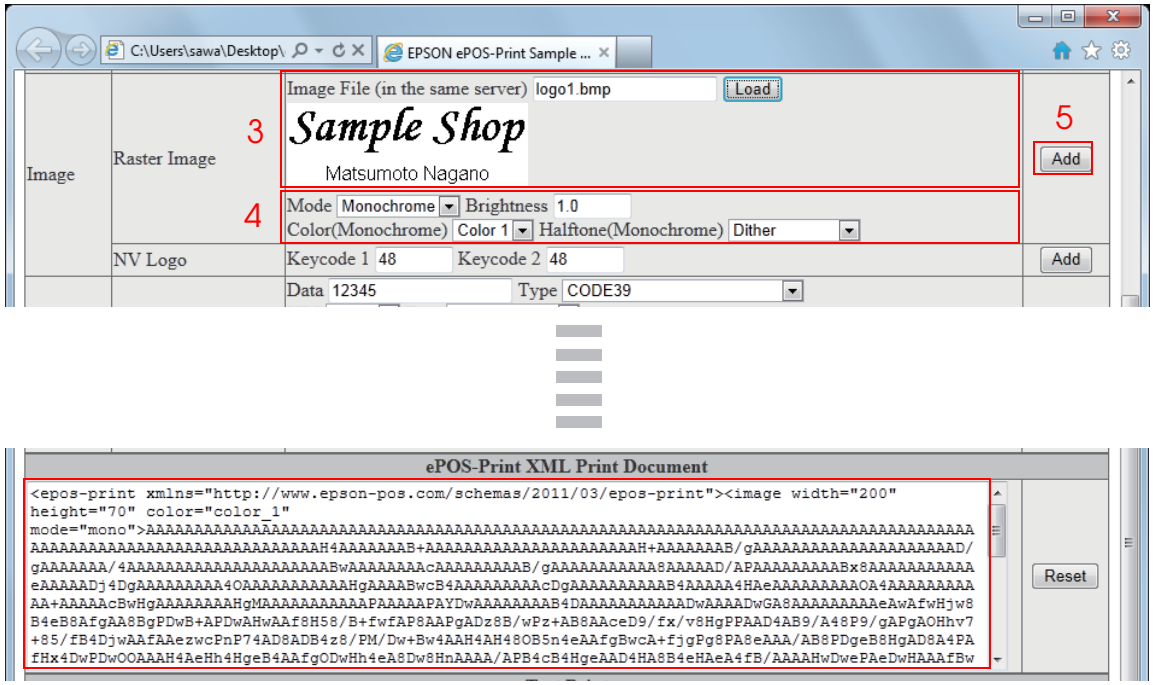
Refer to the following program.

```
//Set the end point address
var url = 'http://192.168.192.168/cgi-bin/epos/service.cgi?devid=local_printer
                                                              &timeout=10000';
//Create an empty print document to check the printer status
var soap = '<?xml version="1.0" encoding="UTF-8"?><s:Envelope xmlns:s="http://
            schemas.xmlsoap.org/soap/envelope/" ><s:Body><epos-print xmlns="http://
            www.epson-pos.com/schemas/2011/03/epos-print"/></s:Body></s:Envelope>';
//Create an XMLHttpRequest object
var xhr = new XMLHttpRequest();
//<Open an XMLHttpRequest object>
xhr.open('POST', url, true);
//<Header settings>
xhr.setRequestHeader('Content-Type', 'text/xml; charset=utf-8');
xhr.setRequestHeader('If-Modified-Since', 'Thu, 01 Jan 1970 00:00:00 GMT');

//Set a response reception callback function for checking
xhr.onreadystatechange = function () {
  if (xhr.readyState == 4){
    if (xhr.status == 200){
      var res = xhr.responseXML;
      var success = res.getElementsByTagName('response')[0].getAttribute('success');
      if (!/^(1|true)$/.test(success)) {
        alert('Success');
      }
    }
  }
};

//Send empty print data
xhr.send(soap);
```

How to Create Graphic Data

To convert image file and raster graphic bit-image data into a base 64 encoded character string, use the ePOSPrintXML generator tool.

How to Use


C:\Users\sawa\Desktop\
EPSON ePOS-Print Sample ...
Image
Raster Image
Image File (in the same server)
logo1.bmp
Load
3
Sample Shop
Matsumoto Nagano
5
Add
4
Mode
Monochrome
Brightness
1.0
Color(Monochrome)
Color 1
Halftone(Monochrome)
Dither
NV Logo
Keycode 1
48
Keycode 2
48
Add
Data
12345
Type
CODE39
6
ePOS-Print XML Print Document
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"><image width="200"
height="70" color="color_1"
mode="mono">AAAA

Printing in Other Environments

This section shows a programming example of printing from a rich internet application.

Microsoft Silverlight (p. 53) and Adobe Flex (p. 54 are used here as examples.

Microsoft Silverlight

```
using System;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Xml.Linq;

namespace SilverlightApplication
{
  public partial class MainPage : UserControl
  {
    private XNamespace soap = "http://schemas.xmlsoap.org/soap/envelope/";
    private XNamespace epos = "http://www.epson-pos.com/schemas/2011/03/epos-print";
    private string address = "http://192.168.192.168/cgi-bin/epos/
                                          service.cgi?devid=local_printer&timeout=10000";
    public MainPage()
    {
      InitializeComponent();
    }

    private void button1_Click(object sender, RoutedEventArgse)
    {
      //Create a print document
      XElement req =
        new XElement(soap + "Envelope",
          new XElement(soap + "Body",
            new XElement(epos + "epos-print",
              new XElement(epos + "barcode", "201234567890",
                new XAttribute("type", "ean13")),
              new XElement(epos + "cut"),
              new XElement(epos + "pulse"))));

      // Send print document
      WebClient client = new WebClient();
      client.Headers["Content-Type"] = "text/xml; charset=utf-8";
      client.UploadStringCompleted +=
                      new UploadStringCompletedEventHandler(client_UploadStringCompleted);
      client.UploadStringAsync(new Uri(address, UriKind.Absolute), req.ToString());
    }

    // Receive a response document
    void client_UploadStringCompleted(object sender, UploadStringCompletedEventArgse)
    {
      if (e.Error != null)
      {
        MessageBox.Show(e.Error.Message);
      }
      else
      {
        // Analyze a response document
        XElement res = XElement.Parse(e.Result);
       var c = from el in res.Descendants(epos + "response") select el.Attribute("success");
        MessageBox.Show(c.First().Value);
      }
    }
}
}
```

Adobe Flex

```
<?xml version="1.0" encoding="utf-8"?>
<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
         xmlns:s="library://ns.adobe.com/flex/spark"
         xmlns:mx="library://ns.adobe.com/flex/mx" minWidth="955" minHeight="600">

  <fx:Script>
    <![CDATA[
      import mx.controls.Alert;

      private var soap:Namespace =
        new Namespace("http://schemas.xmlsoap.org/soap/envelope/");
      private var epos:Namespace =
        new Namespace("http://www.epson-pos.com/schemas/2011/03/epos-print");
      private var address:String =
        "http://192.168.192.168/cgi-bin/epos/service.cgi?devid=local_printer&timeout=10000";

      protected function button1_clickHandler(event:MouseEvent):void
      {
        //Create a print document
        XML.ignoreWhitespace = false;
        XML.prettyPrinting = false;
        var req:XML =
          <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
            <s:Body>
              <epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
                <barcode type="ean13">201234567890</barcode>
                <cut />
                <pulse />
              </epos-print>
            </s:Body>
          </s:Envelope>;

        // Send print document
        var urlRequest:URLRequest = new URLRequest();
        urlRequest.method = URLRequestMethod.POST;
        urlRequest.contentType = "text/xml; charset=utf-8";
        urlRequest.url = address;
        urlRequest.data = req;

        var urlLoader:URLLoader = new URLLoader();
        urlLoader.addEventListener(Event.COMPLETE, completeHandler);
        urlLoader.load(urlRequest);
      }

      // Receive a response document
      private function completeHandler(event:Event):void
      {
        // Analyze a response document
        var res:XML = new XML(event.target.data);
        Alert.show(res.descendants(new QName(epos, "response"))[0].attribute("success"));
      }
    ]]>
  </fx:Script>

  <fx:Declarations>
    <!-- Place a non-visual element here (service, value, object, etc.) -->
  </fx:Declarations>
  <s:Button x="100" y="100" label="Button" click="button1_clickHandler(event)"/>
</s:Application>
```

ePOS-Print XML Reference

This chapter describes Command Reference for ePOS-Print XML print documents.

For differences in specifications according to printer type, refer to Printer specifications (p.91).

Element List

Root element	Element	Function	Standard mode	Page mode	Page
<epos-print>		Request	●	●	p. 56
	<text>	Prints text	●	●	p. 59
	<feed>	Paper feed	●	●	p. 64
	<image>	Prints raster image	●	●	p. 65
	<logo>	Prints NV logo	●	●	p. 67
	<barcode>	Prints barcode	●	●	p. 68
	<symbol>	Prints symbol	●	●	p. 73
	<hline>	Prints horizontal line	●	▲	p. 77
	<vline-begin>	Starts vertical line	●	▲	p. 78
	<vline-end>	Ends vertical line	●	▲	p. 79
	<page>	Page mode	●	▲	p. 80
	<area>	Sets a page mode print area	▲	●	p. 81
	<direction>	Sets a page mode print direction	▲	●	p. 83
	<position>	Sets a page mode print position	▲	●	p. 84
	<line>	Draws a page mode line	▲	●	p. 85
	<rectangle>	Draws a page mode rectangle	▲	●	p. 86
	<cut>	Paper cut	●	▲	p. 87
	<pulse>	Drawer kick-out	●	▲	p. 88
	<sound>	Sounds a buzzer	●	▲	p. 89
	<command>	Inserts a command	●	●	p. 89
<response>		Response	●	●	p. 57

● : Available, ▲ : Not available

Root Element

<epos-print>

An XML document to be sent from a host to a printer.
This document requests the printer to execute the specified function and performs initialization when printing starts.
<epos-print> has elements. Refer to Element List (p. 55).

Example

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print" />
```

<response>

An XML document, a printer returns to a host as a response.
This document sends the printer status after command execution.

The response timing varies depending on the printer status.
- When the power is turned ON: Time set to the TM printer
- When the power is turned OFF: Client timeout time
- When printing, return a response in the following timings:
  * An error occurred: Return an error response before printing.
  * Printing succeeded: Return a success response after printing.

Attribute

- success : (data type xs:boolean)
  Obtains the print result.

Attribute value	Description
true or 1	Printing succeeded.
false or 0	Printing failed.

- code : (data type xs:string)
  Obtains the error code.

Attribute value	Description
EPTR_AUTOMATICAL	An automatically recoverable error occurred
EPTR_COVER_OPEN	A cover open error occurred
EPTR_CUTTER	An autocutter error occurred
EPTR_MECHANICAL	A mechanical error occurred
EPTR_REC_EMPTY	No paper in roll paper end sensor
EPTR_UNRECOVERABLE	An unrecoverable error occurred
SchemaError	The request document contains a syntax error
DeviceNotFound	The printer with the specified device ID does not exist
PrintSystemError	An error occurred on the printing system
EX_BADPORT	An error was detected on the communication port
EX_TIMEOUT	A print timeout occurred

- status : (data type xs:unsignedInt)
  Obtains the printer status.

Attribute value	Description
0x00000001	No response from the TM printer
0x00000002	Printing is successfully completed
0x00000004	Status of the 3rd pin of the drawer kick-out connector = "H"
0x00000008	Offline
0x00000020	The cover is open
0x00000040	Paper is being fed by a paper feed switch operation
0x00000100	Waiting to be brought back online
0x00000200	The paper feed switch is being pressed (ON)
0x00000400	A mechanical error occurred
0x00000800	An autocutter error occurred
0x00002000	An unrecoverable error occurred
0x00004000	An automatically recoverable error occurred
0x00020000	No paper in roll paper near end sensor
0x00080000	No paper in roll paper end sensor
0x01000000	A buzzer is on (only for applicable devices)
0x80000000	The spooler has stopped

Example

When paper runs out while printing and printing fails.

```
<response xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print"
 success="false" code="EPTR_REC_EMPTY" status="252641308" />
```

Element

<text>

Prints a character string. Also, configure the string-related settings such as style, print position, and line feed space.

After printing text, to print content other than text content, execute line feed or paper feed.

In page mode, characters are laid out in the current print position with the reference point being the character baseline dot (Appendix (p.91)).

Entity Reference

To write symbols required for printer control such as horizontal tab, line feed, and the following symbols, use their corresponding entity references shown as follows:

Function (symbol)	Entity Reference
&	&
'	'
>	>
<	<
"	"
Horizontal tab(HT)	or
Line feed (LF)
or

Attribute

- lang : (data type xs:language)
  Specifies the target language.

Attribute value	Description
en(default)	English
ja	Japanese

- font : (data type tns:font)
  Specifies the character font.

Attribute value	Description
FONT_A (default)	Font A
FONT_B	Font B
FONT_C	Font C

When Japanese is specified, characters in Shift-JIS can be printed.

- smooth : (data type xs:boolean)
  Sets smoothing. When smoothing is enabled, the print quality of characters will be improved.

Attribute value	Description
true or 1	Specifies smoothing.
false or 0 (default)	Cancels smoothing

- dw : (data type xs:boolean)
  Sets the double-sized width.

Attribute value	Description
true or 1	Specifies the double-sized width.
false or 0 (default)	Cancels the double-sized width

When the double-sized width attribute and the width attribute are specified in one element at the same time, priority is given to the scale specification for the width attribute.

- dh : (data type xs:boolean)
  Sets the double-sized height.

Attribute value	Description
true or 1	Specifies the double-sized height
false or 0 (default)	Cancels the double-sized height

When the double-sized height attribute and the height attribute are specified in one element at the same time, priority is given to the scale specification for the height attribute.

- width : (data type tns:text-scale)
  Specifies the horizontal scale.

Attribute value	Description
Integer from 1 to 8 (default : 1)	Horizontal scale

When the double-sized width attribute and the width attribute are specified in one element at the same time, priority is given to the scale specification for the width attribute.

- height : (data type tns:text-scale)
  Specifies the vertical scale.

Attribute value	Description
Integer from 1 to 8 (default : 1)	Vertical scale

When the double-sized height attribute and the height attribute are specified in one element at the same time, priority is given to the scale specification for the height attribute.

- reverse : (data type xs:boolean)
  Sets the inversion of black and white parts of characters.

Attribute value	Description
true or 1	Specifies the inversion of black and white parts of characters.
false or 0 (default)	Cancels the inversion of black and white parts of characters.

- ul : (data type xs:boolean)
  Sets underlining.

Attribute value	Description
true or 1	Specifies underlining.
false or 0 (default)	Cancels underlining.

- em : (data type xs:boolean)
  Sets emphasized printing.

Attribute value	Description
true or 1	Specifies emphasized printing.
false or 0 (default)	Cancels emphasized printing.

- color : (data type tns:color)
  Specifies the character color.

Attribute value	Description
none	Characters are not printed.
color_1 (default)	First color
color_2	Second color
color_3	Third color
color_4	Fourth color

- x : (data type xs:unsignedShort Default : 0)
  Specifies the print start position of the text in dots.
  In page mode, this specification is the same as the specification for the horizontal print position by the position element, but the print position in the horizontal direction only will be moved.
- align : (data type tns:align)
  Specifies the alignment.

Attribute value	Description
left (default)	Alignment to the left
center	Alignment to the center
right	Alignment to the right

- For the standard mode, specify the align attribute "when at the start of a line".
- The align attribute set in this element also applies to the align attribute in each of the image, logo, barcode, and symbol elements.

- rotate : (data type xs:boolean)
  Sets rotated printing of text.

Attribute value	Description
true or 1	Specifies rotated printing of text.
false or 0 (default)	Cancels rotated printing of text.

- In page mode, even if rotate is specified, it is disabled.
  For the page mode, set the print direction as right to left (right_to_left) using the direction element to perform 180-degree rotated printing of text.
- The rotate attribute set in this element also applies to the rotate attribute in each of the barcode and symbol elements.

- linespc : (data type xs:unsignedShort Default : 30)
  Specifies the per-line paper feed amount in dots.

Example

To print a character string based on the following settings:

Item	Setting
Language	English
Smoothing	Enable
Align	Center Align
Font	Font A
Double-sizing	Double width and height characters
Underline	Enable

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <text lang="en" />
   <text smooth="true" />
   <text align="center" />
   <text font="font_a" />
   <text dw="true" dh="true" />
   <text ul="true" />
   <text>Hello, World!&#10;</text>
 </epos-print>
```

<feed>

Feeds paper. Specifies the paper feed amount in dots or in lines. When no paper feed amount is specified, performs single-line paper feed (line feed). In addition, the per-line line feed space can be set.

Attribute

- unit : (data type xs:unsignedByte)
  Specifies the paper feed amount in dots.
- line : (data type xs:unsignedByte)
  Specifies the paper feed amount in lines.
- linespc : (data type xs:unsignedByte, Default : 30)
  Specifies the per-line paper feed amount in dots.

- When the per-line paper feed amount is 30 dots, up to 240 lines can be specified.
- The standard mode retains the line feed space separate from the line feed space retained by the page mode. When the linespc attribute is specified in this element, it will affect the linespc attribute in each of the <text> and <feed> elements that follow.

Example

To feed paper 3 lines forward:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <feed line="3" />
</epos-print>
```

To start a new line after printing a character string:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <text>Hello</text><feed />
   <text>World</text><feed />
</epos-print>
```

<image>

Print a raster graphic bit-image. (Data type xs:base64Binary)
Raster graphics refer to data generated by horizontally scanning the pixels of an image from the top left corner of that image as the origin point.
Data is generated based on 1 bit per pixel for two-tone images, and 4 bits per pixel for 16-tone images, starting with the upper bits of byte data. And zero-padding is performed so that scan data per line can be treated on a byte basis.

To print a raster image at high speed, specify "left" for the align attribute and specify a multiple of 8 that does not exceed the printer paper width for the width attribute value.

- In page mode, a raster image is laid out in the current print position with the reference point being its bottom left dot. The print position will not move.
- In page mode, set the print position so that a raster image does not extend beyond the print area.
- In page mode, you can not print on multiple tones.
- When multiple tones are set for raster images, intermitting printing may occur because the amount of data to print increases and white stripes may appear in the print result.
- The scan quality of barcodes/two-dimensional codes printed as multiple-tone raster images cannot be guaranteed. Print them as two-tone images.
- To create raster graphic bit-images, use the ePOS-Print XML generator tool or your own application. Note the following when creating raster graphic images using your own application.
  Two-tone image: Specify a multiple of 8 for the image width or fill the missing bits with zeros.
  16-tone image: Specify a multiple of 2 for the image width or fill the missing bits with zeros.

Attribute

- width : (Required attribute, data type xs:unsignedShort)
  Specifies the image width in dots.
- height : (Required attribute, data type xs:unsignedShort)
  Specifies the image height in dots.
- color : (Optional attribute, data type tns:color)
  Specifies the character color.

Attribute value	Description
none	Characters are not printed.
color_1 (default)	First color
color_2	Second color
color_3	Third color
color_4	Fourth color

- align : (data type tns:align)
  Specifies the print position.

Attribute value	Description
left (default)	Alignment to the left
center	Alignment to the center
right	Alignment to the right

The align attribute set in this element also applies to the align attribute in each of the text, logo, barcode, and symbol elements.

- mode : (Optional attribute, data type tns:image-mode)
  Specifies the color mode.

Attribute value	Description
mono (default)	Monochrome (two-tone)
gray16	Multiple tones (16-tone)

Example

To print an 8 dot wide, 8 dot high filled-in raster image:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <image width="8" height="8">//////////8=</image>
</epos-print>
```

<logo>

Prints a logo registered in the NV memory of the printer.

- Using TM-T88V Utility or logo registration utility (TMFLogo), register a logo in the printer in advance.
- In page mode, a logo is laid out in the current print position with the reference point being its bottom left dot.

Attribute

- key1 : (Required attribute, data type xs:unsignedByte)
  Specifies the value of the key code 1 set at the time of NV logo registration.
- key2 : (Required attribute, data type xs:unsignedByte)
  Specifies the value of the key code 2 set at the time of NV logo registration. Be sure to specify this value.
- align : (data type tns:align)
  Specifies the print position.

Attribute value	Description
left (default)	Alignment to the left
center	Alignment to the center
right	Alignment to the right

The align attribute set in this element also applies to the align attribute in each of the text, image, barcode, and symbol elements.

Example

To print a NV logo with key1 registered as 48 and key2 registered as 48:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <logo key1="48" key2="48" />
</epos-print>
```

<barcode>

Prints a barcode.
Specifies the barcode data using a character string. (Data type xs:string)

- A barcode will not be printed if settings not compliant with the barcode standard are made, or if the barcode becomes larger than the printer's print area. And an error response will not be returned, either.
- In page mode, a barcode is laid out in the current print position with the reference point being its bottom left dot (except for HRI).

Barcode Type

The following barcodes can be specified:

Barcode type	Barcode data specification procedure
UPC-A	When an 11-digit number is specified, a check digit is automatically added. When a 12-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.
UPC-E	Specify 0 as the first digit. Specify the manufacturer code in the digits 2 to 6. Specify (right-align) the item code in the digits 7 to 11. The number of item code digits varies depending on the manufacturer code. Specify 0s in empty digits. When an 11-digit number is specified, a check digit is automatically added. When a 12-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.
EAN13 JAN13	When an 12-digit number is specified, a check digit is automatically added. When a 13-digit number is specified, the 12th digit is processed as a check digit but the check digit is not validated.
EAN8 JAN8	When a 7-digit number is specified, a check digit is automatically added. When an 8-digit number is specified, the 8th digit is processed as a check digit but the check digit is not validated.
CODE39	When the first character is *, the character is processed as the start character. In other cases, a start character is automatically added.
ITF	Start and stop codes are automatically added. Check digits are not added or validated.
CODABAR	Specify a start character (A to D, a to d). Specify a stop character (A to D, a to d). Check digits are not added or validated.
CODE93	Start and stop characters are automatically added. A check digit is automatically calculated and added.

Barcode type	Barcode data specification procedure
CODE128	Specify a start character (CODE A, CODE B, CODE C). A stop character is automatically added. A check digit is automatically calculated and added. To encode each of the following characters, specify two characters starting with the character "{": FNC1: {1 FNC2: {2 FNC3: {3 FNC4: {4 CODE A: {A CODE B: {B CODE C: {C SHIFT: {S {: {{
GS1-128	A start character, FNC1, a check digit, and a stop character are automatically added. To automatically calculate and add a check digit for an application identifier (AI) and the subsequent data, specify the character "*" in the position of the check digit. You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data. You can insert spaces between an application identifier (AI) and data. The spaces are used as HRI print characters and are not encoded as data. To encode each of the following characters, specify two characters starting with the character "{": FNC1: {1 FNC3: {3 (: {( ): {) *: {* {: {{
GS1 DataBar Omnidirectional GS1 DataBar Truncated GS1 DataBar Limited	Specify a 13-digit global trade item number (GTIN) not including an application identifier (AI) or a check digit.

Barcode type	Barcode data specification procedure
BARCODE_GS1_ DATABAR_EXPANDED	You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data. To encode each of the following characters, specify two characters starting with the character "{": FNC1: {1 (: {( ): {)

To specify binary data that cannot be represented by character strings, use the following escape sequences.

String	Description
\xnn	Control code
\\	Back slash

Attribute

- type : (Required attribute, data typetns:barcode-type)
  Specifies the barcode type.

Attribute value	Description
upc_a	UPC-A
upc_e	UPC-E
ean13	EAN13
jan13	JAN13
ean8	EAN8
jan8	JAN8
code39	CODE39
itf	ITF
codabar	CODABAR
code93	CODE93
code128	CODE128
gs1_128	GS1-128
gs1_databar_omnidirectional	GS1 DataBar Omnidirectional
gs1_databar_truncated	GS1 DataBar Truncated
gs1_databar_limited	GS1 DataBar Limited
gs1_databar_expanded	GS1 Databar Expanded

- hri : (data type tns:barcode-hri)
  Specifies the HRI position.

Attribute value	Description
none (default)	HRI not printed
above	Above the bar code
below	Below the bar code
both	Both above and below the bar code

- font : (data type tns:font)
  Specifies the HRI font.

Attribute value	Description
font_a (default)	Font A
font_b	Font B
font_c	Font C

- width : (data type xs:unsignedByte Default "3")
  Specifies the width of each module in dots. Specifies an integer from 2 to 6.
- height : (data type xs:unsignedByte Default "162")
  Specifies the barcode height in dots.
- align : (data type tns:align)
  Specifies the print position.

Attribute value	Description
left (default)	Alignment to the left
center	Alignment to the center
right	Alignment to the right

The align attribute set in this element also applies to the align attribute in each of the text, image, logo, and symbol elements.

- rotate : (data type xs:boolean)
  Specifies the rotate printing.

Attribute value	Description
true or 1	Specifies rotated printing of text.
false or 0 (default)	Cancels rotated printing of text.

The rotate attribute set in this element also applies to the rotate attribute in each of the text and symbol elements.

Example

To print barcodes:

```
epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <barcode type="upc_a" width="2" height="64" hri="below">01234567890</
        barcode>
  <barcode type="upc_e">01234500005</barcode>
  <barcode type="ean13">201234567890</barcode>
  <barcode type="jan13">201234567890</barcode>
  <barcode type="ean8">2012345</barcode>
  <barcode type="jan8">2012345</barcode>
  <barcode type="code39">ABCDE</barcode>
  <barcode type="itf">012345</barcode>
  <barcode type="codabar">A012345A</barcode>
  <barcode type="code93">ABCDE</barcode>
  <barcode type="code128">{Babcde</barcode>
  <barcode type="gs1_128">(01)201234567890*</barcode>
  <barcode type="gs1_databar_omnidirectional">0201234567890</barcode>
  <barcode type="gs1_databar_truncated">0201234567890</barcode>
  <barcode type="gs1_databar_limited">0201234567890</barcode>
  <barcode type="gs1_databar_expanded">(01)2012345678903</barcode>
</epos-print>
```

<symbol>

Prints a two-dimensional symbol.
Specifies the two-dimensional symbol using a character string. (Data type xs:string)

- A two-dimensional symbol will not be printed if settings not compliant with the two-dimensional symbol standard are made, or if the two-dimensional symbol becomes larger than the printer's print area. And an error response will not be returned, either.
- In standard mode, a more than 831 dot high two-dimensional code cannot be printed.
- In page mode, a two-dimensional symbol is laid out in the current print position with the reference point being its top left dot.

Two-Dimensional Symbol Type

The following two-dimensional symbols can be specified:

Type	Description
PDF417	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. The data area can contain up to 928 code words in a maximum of 90 rows, each of which can contain up to 30 code words.
QR Code	Convert the character string to the string in Shift-JIS, apply the escape sequence, and then encode the string based on the data type as shown below. Number: 0 to 9 Alphanumeric character: 0 to 9, A to Z, space, $, %, *, +, -, ., /, : Kanji character: Shift-JIS value 8-bit, byte data: 0x00 to 0xff
MaxiCode	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. In Modes 2 and 3, when the first piece of data is ()>\ x1e01\x1dyy (where yy is a two-digit number), this is processed as the message header, and the subsequent data is processed as the primary message. In other cases, from the first piece of data, data is processed as the primary message. In Mode 2, specify the primary message in the following format: Postal code (1- to 9-digit number) GS:(\x1d) ISO country code (1- to 3-digit number) GS:(\x1d) Service class code (1- to 3-digit number) In Mode 3, specify the primary message in the following format: Postal code (1 to 6 pieces of data convertible by Code Set A) GS:(\x1d) ISO country code (1- to 3-digit number) GS:(\x1d) Service class code (1- to 3-digit number)

Type	Description
GS1 DataBar Stacked	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. Specify a 13-digit global trade item number (GTIN) not including an application identifier (AI) or a check digit.
GS1 DataBar Stacked Omnidirectional	
GS1 DataBar Expanded Stacked	Convert the character string to the string in UTF-8, apply the escape sequence, and then encode the string. You can enclose an application identifier (AI) in parentheses. The parentheses are used as HRI print characters and are not encoded as data. To encode each of the following characters, specify two characters starting with the character "{": FNC1: {1 (: {( ): {)

To specify binary data that cannot be represented by character strings, use the following escape sequences.

String	Description
\xnn	Control code
\\	Back slash

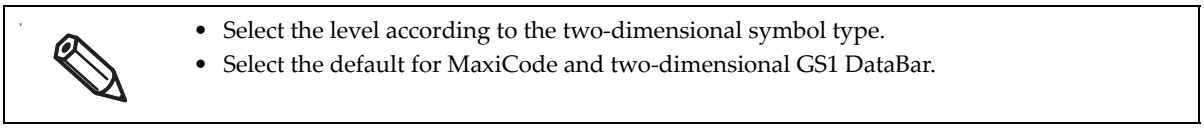
Attribute

- type : (Required attribute, data type tns:symbol-type)
  Specifies the two-dimensional symbol type.

Attribute value	Type
pdf417_standard	Standard PDF417
pdf417_truncated	Truncated PDF417
qrcode_model_1	QR Code Model 1
qrcode_model_2	QR Code Model 2
maxicode_mode_2	MaxiCode Mode 2
maxicode_mode_3	MaxiCode Mode 3
maxicode_mode_4	MaxiCode Mode 4
maxicode_mode_5	MaxiCode Mode 5
maxicode_mode_6	MaxiCode Mode 6
gs1_databar_stacked	GS1 DataBar Stacked
gs1_databar_stacked_omnidirectional	GS1 DataBar Stacked Omnidirectional
gs1_databar_expanded_stacked	GS1 DataBar Expanded Stacked

- level : (data type tns:symbol-level)
  Specifies the error correction level.

Attribute value	Description
level_0	PDF417 error correction level 0
level_1 (default)	PDF417 error correction level 1
level_2	PDF417 error correction level 2
level_3	PDF417 error correction level 3
level_4	PDF417 error correction level 4
level_5	PDF417 error correction level 5
level_6	PDF417 error correction level 6
level_7	PDF417 error correction level 7
level_8	PDF417 error correction level 8
level_l	QR Code error correction level L
level_m	QR Code error correction level M
level_q	QR Code error correction level Q
level_h	QR Code error correction level H
default	Default level



- Select the level according to the two-dimensional symbol type.
- Select the default for MaxiCode and two-dimensional GS1 DataBar.

- width : (data type xs:unsignedByte)
  Specifies the width of each module in dots.

Two-dimensional symbol	Valid value range	Default
PDF417	2 to 8	width="3"
QR Code	1 to 16	width="3"
MaxiCode	Ignored	
GS1 DataBar	2 to 8	width="2"

- height : (data type xs:unsignedByte)
  Specifies the height of each module in dots.

Two-dimensional symbol	Valid value range	Default
PDF417	2 to 8	height="3"
QR Code	Ignored	
MaxiCode		
GS1 DataBar		

- size : (data type xs:unsignedShort)

Two-dimensional symbol	Default	Description
PDF417	size="0"	Specifies the number of code words for each row
QR Code	Ignored	
MaxiCode		
GS1 DataBar	Varies depending on the model	Specifies the maximum width for the barcode

- align : (data type tns:align)
  Specifies the print position.

Attribute value	Description
left (default)	Alignment to the left
center	Alignment to the center
right	Alignment to the right

The align attribute set in this element also applies to the align attribute in each of the text, image, logo, and barcode elements.

- rotate : (data type xs:boolean)
  Specifies the rotate printing.

Attribute value	Description
true or 1	Specifies rotated printing of text.
false or 0 (default)	Cancels rotated printing of text.

The rotate attribute set in this element also applies to the rotate attribute in each of the text and barcode elements.

Example

To print two-dimensional symbols:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <symbol type="pdf417_standard">ABCDE</symbol>
  <symbol type="qrcode_model_2" level="level_q">ABCDE</symbol>
  <symbol type="maxicode_mode_2">908063840\x1d850\x1d001\x1d\x04</symbol>
  <symbol type="gs1_databar_stacked">0201234567890</symbol>
  <symbol type="gs1_databar_stacked_omnidirectional">0201234567890</symbol>
  <symbol type="gs1_databar_expanded_stacked">(01)02012345678903</symbol>
</epos-print>
```

<hline>

Draws horizontal lines.

> Not available in page mode.

Attribute

- x1 : (Required attribute, data type xs:unsignedShort)
  Specifies the draw start position of the horizontal line (in dots).
- x2 : (Required attribute, data type xs:unsignedShort)
  Specifies the draw end position of the horizontal line (in dots).
- style : (Optional attribute, data type tns:line-style)
  Specifies the line type.

Attribute value	Description
thin(default)	Solid line: Thin
medium	Solid line: Medium
thick	Solid line: Thick
thin_double	Double line: Thin
medium_double	Double line: Medium
thick_double	Double line: Thick

Example

To draw double lines in the following positions:

- From the left end: 100th dot to 200th dot
- From the left end: 400th dot to 500th dot

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <hline x1="100" x2="200" style="thin_double" />
  <hline x1="400" x2="500" style="thin_double" />
</epos-print>
```

<vline-begin>

Starts to draw vertical lines.

	Not available in page mode.
	A vertical line is drawn up to the end position specified by the vline-end element (p. 79). Use this element together with the vline-end element.

Attribute

- x : (Required attribute, data type xs:unsignedShort)
  Specifies the start position of the horizontal line (in dots).
- style : (Optional attribute, data type tns:line-style)
  Specifies the line type.

Attribute value	Description
thin(default)	Solid line: Thin
medium	Solid line: Medium
thick	Solid line: Thick
thin_double	Double line: Thin
medium_double	Double line: Medium
thick_double	Double line: Thick

Example

To draw a thin line from 100th to 200th dot from the left end.

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
    <vline-begin x="100" />
    <vline-begin x="200" />
    <feed unit="100" />
    <vline-end x="100" />
    <vline-end x="200" />
</epos-print>
```

<vline-end>

Ends to draw vertical lines.

Not available in page mode.

Use this element together with the vline-begin element (p. 78).

Attribute

- x : (Required attribute, data type xs:unsignedShort)
  Specifies the end position of the horizontal line (in dots).
- style : (Optional attribute, data type tns:line-style)
  Specifies the line type.

Attribute value	Description
thin(default)	Solid line: Thin
medium	Solid line: Medium
thick	Solid line: Thick
thin_double	Double line: Thin
medium_double	Double line: Medium
thick_double	Double line: Thick

Example

To draw a thin line from 100th to 200th dot from the left end.

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <vline-begin x="100" />
   <vline-begin x="200" />
   <feed unit="100" />
   <vline-end x="100" />
   <vline-end x="200" />
</epos-print>
```

<page>

Changes the mode from standard mode to page mode.

Elements via <page>

Via <page>, the following elements can be used.

Element	Description
<text>	Prints text
<feed>	Paper feed
<image>	Prints a raster image
<logo>	Prints a NV logo
<barcode>	Prints a barcode
<symbol>	Prints a two-dimensional symbol
<area>	Sets a print area
<direction>	Sets a print direction
<position>	Sets a print position
<line>	Draws a line
<rectangle>	Draws a rectangle
<command>	Inserts a command

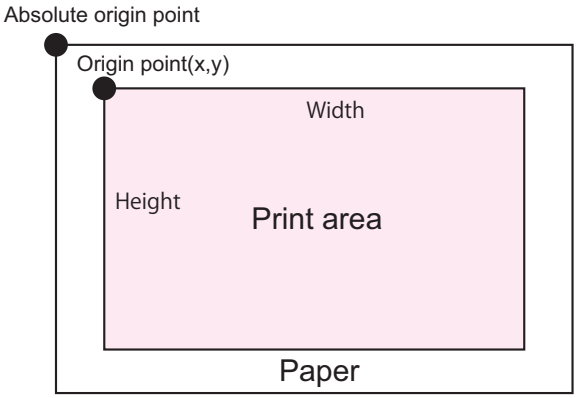
Example

To print "ABCDE" in page mode:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <page>
   <text>ABCDE</text>
   </page>
</epos-print>
```

<area>

Specifies the page mode print area.
Specifies the origin point, width, and height based on the absolute origin point to set the print area. The absolute origin point is a dot diagonally above the top left corner of the printable area.



⚠	Specify the print area according to the print content. If print data extends beyond the print area, the print result is such that the whole data cannot be printed completely.
✎	Use this element inside the page element.

Attribute

- x : (Required attribute, data type xs:unsignedShort Default "0")
  Specifies the origin of the horizontal axis (in dots).
- y : (Required attribute, data type xs:unsignedShort Default "0")
  Specifies the origin of the vertical axis (in dots).
- width : (Required attribute, data type xs:unsignedShort Default :Differ according to each model)
  Specifies the width of the print area (in dots).
- height : (Required attribute data type xs:unsignedShort Default :Differ according to each model)
  Specifies the height of the print area (in dots).

⚠	Determine the width and height of the print area according to the print direction setting. Otherwise, the print data might not be printed completely.

Example

To print the characters "ABCDE" by specifying the print area as origin (100, 50), width: 200 dots, and height: 30 dots:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <page>
  <area x="100" y="50" width="200" height="30" />
  <text>ABCDE</text>
  </page>
</epos-print>
```

<direction>

Specifies the page mode print direction.
Specifies the print direction and rotates the print area. Following the rotation of the print area, the start point of the print area moves.

Use this element inside the page element.

Attribute

- dir : (Required attribute, data type tns:page-dir)
  Specifies the rotation direction.

Attribute value	Description
left_to_right (default)	Left to right (No rotation.Data is printed from the top left corner to the right.)
bottom_to_top	Bottom to top (Counterclockwise rotation by 90 degrees. Data is printed from the bottom left corner to the top.)
right_to_left	Right to left (Rotation by 180 degrees.Data is printed from the bottom right corner to the left.)
top_to_bottom	Top to bottom (Clockwise rotation by 90 degrees. Data is printed from the top right corner to the bottom.)

Example

To print the characters "ABCDE" by rotating them 90 degrees clockwise:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <page>
   <direction dir="top_to_bottom" />
   <text>ABCDE</text>
   </page>
</epos-print>
```

<position>

Specifies the page mode print position.
Specifies the print position based on the start point of the print area. The start point of the print area moves by following the rotation of the print area.

Use this element inside the page element.

Attribute

- x : (Required attribute, data type xs:unsignedShort Default "0")
  Specifies the origin of the horizontal axis (in dots).
- y : (Required attribute, data type xs:unsignedShort Default "21")
  Specifies the origin of the vertical axis (in dots).

Specify the print start position (coordinates) according to the content to be printed. Refer to the following.

* To print a character string:
  Specify the left end of the baseline for the first character. This can be omitted for left-aligned printing of standard-sized characters. To print double-sized height characters, specify a value equal to or greater than 42 for y.
* To print a barcode:
  Specify the bottom left of the symbol. And specify the barcode height for y.
* To print a graphic/logo:
  Specify the bottom left of the graphic data. And specify the graphic data height for y.
* To print a two-dimensional symbol:
  Specify the top left of the symbol. This can be omitted when printing from the top left.

Example

To specify (50,30) for the print start position in the area specified by the addPageArea method and print the characters "ABCDE":

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <page>
  <area x="100" y="50" width="200" height="100" />
  <position x="50" y="30" />
  <text>ABCDE</text>
  </page>
</epos-print>
```

<line>

Draws a line in page mode.

Diagonal lines cannot be drawn.

Use this element inside the page element.

Attribute

- x1: (Required attribute, data type xs:unsignedShort)
  Specifies the drawing start position of the horizontal line (in dots).
- y1 : (Required attribute, data type xs:unsignedShort)
  Specifies the drawing start position of the vertical line (in dots).
- x2 : (Required attribute, data type xs:unsignedShort)
  Specifies the drawing end position of the horizontal line (in dots).
- y2 : (Required attribute, data type xs:unsignedShort)
  Specifies the drawing end position of the vertical line (in dots).
- style : (data type tns:line-style)
  Specifies the line type.

Attribute value	Description
thin	Solid line: Thin
medium	Solid line: Medium
thick	Solid line: Thick
thin_double	Double line: Thin
medium_double	Double line: Medium
thick_double	Double line: Thick

Example

To draw a thin solid line between the start position (100, 0) and the end position (500, 0):

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <page>
   <line x1="100" y1="0" x2="500" y2="0" style="thin" />
   </page>
</epos-print>
```

<rectangle>

Draws a rectangle in page mode.

- Use this element inside the page element.
- Not available in standard mode. Use the hline, vline-begin, and vline-end elements instead.

Attribute

- x1: (Required attribute, data type xs:unsignedShort)
  Specifies the drawing start position of the horizontal line (in dots).
- y1 : (Required attribute, data type xs:unsignedShort)
  Specifies the drawing start position of the vertical line (in dots).
- x2 : (Required attribute, data type xs:unsignedShort)
  Specifies the drawing end position of the horizontal line (in dots).
- y2 : (Required attribute, data type xs:unsignedShort)
  Specifies the drawing end position of the vertical line (in dots).
- style : (data type tns:line-style)
  Specifies the line type.

Attribute value	Description
thin	Solid line: Thin
medium	Solid line: Medium
thick	Solid line: Thick
thin_double	Double line: Thin
medium_double	Double line: Medium
thick_double	Double line: Thick

Example

To draw a rectangle with a thin double line, with the start position (100, 0) and the end position (500, 200) as its vertexes:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
  <page>
  <rectangle x1="100" y1="0" x2="500" y2="200" style="thin-double" />
  </page>
</epos-print>
```

<cut>

Sets paper cut.

Available in standard mode. Not available in page mode.

Execute "when at the top of a line". When the paper is cut, the state turns to "when at the top of a line".

Attribute

- type : (data type tns:cut-type)
  Specifies the paper cut type.

Attribute value	Description
no_feed	Cut without feeding (The paper is cut without being fed.)
feed	Feed cut (The paper is fed to the cut position and then is cut.)
reserve	Cut reservation (Printing continues until the cut position is reached, at which the paper is cut.)

Example

To perform feed cut operation:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <cut type="feed" />
</epos-print>
```

<pulse>

Outputs signals to the drawer kick connector.
Depending on the model, a buzzer can be sounded.

- Not available in page mode.
- The drawer and the buzzer cannot be used together.

Attribute

- drawer : (data type tns:pulse-drawer)
  Specifies the drawer kick connector.

Attribute value	Description
drawer_1	Pin 2 of the drawer kick-out connector
drawer_2	Pin 5 of the drawer kick-out connector

- time : (data type tns:pulse-time)
  Specifies the ON time of the drawer kick signal.

Attribute value	Description
pulse_100	100 ms
pulse_200	200 ms
pulse_300	300 ms
pulse_400	400 ms
pulse_500	500 ms

Example

To send a 100msec pulse signal to the pin 2 of the drawer kick connector:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <pulse drawer="drawer_1" time="pulse_100" />
</epos-print>
```

<sound>

Sounds a buzzer.

- Not available in page mode.
- The buzzer function and the drawer cannot be used together.
- This element is not available if the printer is not provided with the buzzer.

Attribute

- pattern : (data type tns:sound-pattern)
  Specifies the buzzer pattern.

Attribute value	Description
none	Stop
pattern_a	Pattern A
pattern_b	Pattern B
pattern_c	Pattern C
pattern_d	Pattern D
pattern_e	Pattern E
error	Error sound pattern
paper_end	Pattern when there is no paper

- repeat : (data type xs:unsignedByte)
  Specifies the number of repeats.

Attribute value	Description
0	The buzzer does not stop.
1 to 255	Number of repeats

Example

To repeat the sound pattern A three times:

```
<epos-print xmlns="http://www.epson-pos.com/schemas/2011/03/epos-print">
   <sound pattern="pattern_a" repeat="3" />
</epos-print>
```

<command>

Insert the ESC / POS command.
ESC / POS command is specified in hexadecimal encoding. (data type xs:hexBinary)

ESC / POS command is not open to the public.

Appendix

Printer specifications

TM-T88V-i

		80mm
Interface		Ethernet, Wireless LAN
Resolution		180 dpi x 180 dpi (W x H)
Print Width		512 dots
Font		For more information about what character codes can be printed, refer to the user's manual that came with the printer.
Characters in a Line	Font A	ANK: 42 characters
	Font B	ANK: 56 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)
	Font B	ANK: 9 dots x 17 dots (W x H)
Character Baseline	Font A	At the 21st dot from the top of the character
	Font B	At the 16th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Raster Image <image>		Monochrome image, Gray scale image
Logo <logo>		Monochrome image, Gray scale image
Bar Code <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128,GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded
Two-Dimensional Code <symbol>		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked (Composite Symbology not supported)
Ruled Line <hline>, <vline-xxx>		Not supported

		80mm
Page Mode Default Area		512 dots x 831 dots (W x H)
Page Mode Maximum Area		512 dots x 1662 dots (W x H)
Page Mode	Line<line>	Not supported
	Rectangle <rectangle>	
Paper Cut <cut>		Cut, Feed cut
Drawer Kick-Out <pulse>		Supported
Buzzer <sound>		Optional
Command <command>		Supported

TM-T88V

		80mm
Interface		Ethernet, Wireless LAN
Resolution		180 dpi x 180 dpi (W x H)
Print Width		512 dots
Font		For more information about what character codes can be printed, refer to the user's manual that came with the printer.
Characters in a Line	Font A	ANK: 42 characters
	Font B	ANK: 56 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)
	Font B	ANK: 9 dots x 17 dots (W x H)
Character Baseline	Font A	At the 21st dot from the top of the character
	Font B	At the 16th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Raster Image <image>		Monochrome image, Gray scale image
Logo <logo>		Monochrome image, Gray scale image
Bar Code <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128, GS1-128,GS1 DataBar Omnidirectional, GS1 DataBar Truncated, GS1 DataBar Limited, GS1 Databar Expanded
Two-Dimensional Code <symbol>		PDF417, QR Code, MaxiCode, GS1 DataBar Stacked, GS1 DataBar Stacked Omnidirectional, GS1 DataBar Expanded Stacked (Composite Symbology not supported)
Ruled Line <hline>, <vline-xxx>		Not supported
Page Mode Default Area		512 dots x 831 dots (W x H)
Page Mode Maximum Area		512 dots x 1662 dots (W x H)
Page Mode	Line<line>	Not supported
	Rectangle <rectangle>	

	80mm
Paper Cut <cut>	Cut, Feed cut
Drawer Kick-Out <pulse>	Supported
Buzzer <sound>	Optional
Command <command>	Supported

TM-T70-i

		80mm
Interface		Ethernet, Wireless LAN
Resolution		180 dpi x 180 dpi (W x H)
Print Width		512 dots
Font		For more information about what character codes can be printed, refer to the user's manual that came with the printer.
Characters in a Line	Font A	ANK: 42 characters
	Font B	ANK: 56 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)
	Font B	ANK: 9 dots x 17 dots (W x H)
Character Baseline	Font A	At the 21st dot from the top of the chara
	Font B	At the 16th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Raster Image<image>		Monochrome image
Logo<logo>		Monochrome image
Bar Code <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128
Two-Dimensional Code <symbol>		PDF417, QR Code
Ruled Line <hline>, <vline-xxx>		Not supported
Page Mode Default Area		512 dots x 1662 dots (W x H)
Page Mode Maximum Area		512 dots x 1662 dots (W x H)mv
Page Mode	Line<line>	Not supported
	Rectangle <rectangle>	
Paper Cut <cut>		Cut, Feed cut
Drawer Kick-Out <pulse>		Supported
Buzzer <sound>		Not supported
Command <command>		Supported

TM-T70

		80mm
Interface		Ethernet, Wireless LAN
Resolution		180 dpi x 180 dpi (W x H)
Print Width		512 dots
Font		For more information about what character codes can be printed, refer to the user's manual that came with the printer.
Characters in a Line	Font A	ANK: 42 characters
	Font B	ANK: 56 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)
	Font B	ANK: 9 dots x 17 dots (W x H)
Character Baseline	Font A	At the 21st dot from the top of the chara
	Font B	At the 16th dot from the top of the character
Default Line Feed Space		30 dots
Color Specification		First color
Raster Image<image>		Monochrome image
Logo<logo>		Monochrome image
Bar Code <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128
Two-Dimensional Code <symbol>		PDF417, QR Code
Ruled Line <hline>, <vline-xxx>		Not supported
Page Mode Default Area		512 dots x 1662 dots (W x H)
Page Mode Maximum Area		512 dots x 1662 dots (W x H)mv
Page Mode	Line<line>	Not supported
	Rectangle <rectangle>	
Paper Cut <cut>		Cut, Feed cut
Drawer Kick-Out <pulse>		Supported
Buzzer <sound>		Not supported
Command <command>		Supported

TM-T90

		58mm	60mm	80mm
Interface		Ethernet, Wireless LAN		
Resolution		180 dpi x 180 dpi (W x H)		
Print Width		360 dots	384 dots	512 dots
Font		For more information about what character codes can be printed, refer to the user's manual that came with the printer.		
Characters in a Line	Font A	ANK: 30 characters,	ANK: 32 characters	ANK: 42 characters
	Font B	ANK: 40 characters	ANK: 42 characters	ANK: 56 characters
Character Size	Font A	ANK: 12 dots x 24 dots (W x H)		
	Font B	ANK: 9 dots x 17 dots (W x H)		
Character Baseline	Font A	At the 21st dot from the top of the character		
	Font B	At the 16 th dot from the top of the character		
Default Line Feed Space		30 dots		
Color Specification		First color Second color, Second color (when two-color printing is set)		
Raster Image<image>		Monochrome image, Two color image		
Logo<logo>		Monochrome image, Two color image (To perform two-color printing, change the settings of the printer using the memory switch setting utility.)		
Bar Code <barcode>		UPC-A, UPC-E, EAN13, JAN13, EAN8, JAN8, CODE39, ITF, CODABAR, CODE93, CODE128		
Two-Dimensional Code <symbol>		PDF417		
Ruled Line <hline>, <vline-xxx>		Not supported		
Page Mode Default Area		360 dots x 831 dots (W x H)	384 dots x 831 dots (W x H)	512 dots x 831 dots (W x H)
	when two-color printing is set	360 dots x 415 dots (W x H)	384 dots x 415 dots (W x H)	512 dots x 415 dots (W x H)

		58mm	60mm	80mm
Page Mode Maximum Area		360 dots x 1662 dots (W x H)	384 dots x 1662 dots (W x H)	512 dots x 1662 dots (W x H)
when two-color printing is set		360 dots x 831 dots (W x H)	384 dots x 831 dots (W x H)	512 dots x 831 dots (W x H)
Page Mode	Line<line>	Not supported		
	Rectangle <rectangle>			
Paper Cut <cut>		Cut, Feed cut		
Drawer Kick-Out <pulse>		Supported		
Buzzer <sound>		Supported via Drawer Kick-Out		
Command <command>		Supported		